

JÉSSICA FONSECA BRUM DA SILVA

ESTRUTURAÇÃO DE DADOS OCEANOGRÁFICOS
UTILIZANDO UM BANCO DE DADOS RELACIONAL: UM
ESTUDO DE CASO COM O MODELO OCCAM

RIO GRANDE – RS
2016

UNIVERSIDADE FEDERAL DO RIO GRANDE
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM COMPUTACIONAL

JÉSSICA FONSECA BRUM DA SILVA

ESTRUTURAÇÃO DE DADOS OCEANOGRÁFICOS
UTILIZANDO UM BANCO DE DADOS RELACIONAL: UM
ESTUDO DE CASO COM O MODELO OCCAM

Dissertação submetida à Universidade
Federal do Rio Grande como parte dos
requisitos para a obtenção do grau de
Mestre em Modelagem Computacional.

Área de Concentração: Ciências Exatas e da
Terra

Orientadora: Profa. Dra. Diana F. Adamatti
Co-Orientadora: Profa. Dra. Elisa Helena
Fernandes

RIO GRANDE – RS
2016

Ficha catalográfica

S586e Silva, Jéssica Fonseca Brum da.
Estruturação de dados oceanográficos utilizando um banco de dados relacional: um estudo de caso com o modelo OCCAM / Jéssica Fonseca Brum da Silva. – 2016.
104 f.

Dissertação (mestrado) – Universidade Federal do Rio Grande – FURG, Programa de Pós-graduação em Modelagem Computacional, Rio Grande/RS, 2016.

Orientadora: Dr^a. Diana Francisca Adamatti.

Coorientadora: Dr^a. Elisa Helena Fernandes.

1. Banco de dados relacional 2. Dados oceanográficos 3. Modelo OCCAM I. Adamatti, Diana Francisca II. Fernandes, Elisa Helena III. Título.

CDU 004.65

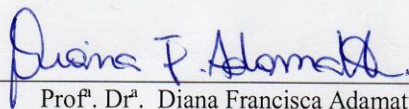
JÉSSICA FONSECA BRUM DA SILVA

**“Estruturação de Dados Oceanográficos Utilizando um Modelo Entidade-Relacionamento:
Um Estudo de Caso com o Modelo OCCAM”**

Dissertação apresentada ao Programa de Pós
Graduação em Modelagem Computacional da
Universidade Federal do Rio Grande - FURG,
como requisito parcial para obtenção do Grau
de Mestre. Área concentração: Modelagem
Computacional.

Aprovada em

BANCA EXAMINADORA



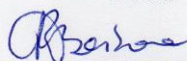
Prof.^a Dr.^a Diana Francisca Adamatti
Orientadora – FURG



Prof.^a Dr.^a Elisa Helena Leão Fernandes
Co-orientadora – FURG



Prof. Dr. Leonardo Ramos Emmendorfer
FURG



Prof. Dr. Raquel de Miranda Barbosa
IFRS

Rio Grande - RS
2016

Posso todas as coisas naquele que me fortalece.

Filipenses 4.13

A Deus, por todas as suas bênçãos em minha
vida e por permitir que este trabalho
fosse concluído.

Agradecimentos

Agradeço primeiramente a Deus por permitir que eu concluísse este trabalho.

Agradeço muito aos meus pais Marcos e Jucelda pelo amor, esforço para nunca faltar nada e pelas inúmeras vezes em que abdicaram os seus sonhos para realizar os meus, abrindo mão de suas vontades para realizar meus caprichos. Obrigada, meus pais, por nunca deixar de acreditar que eu conseguiria.

A meu marido Eliel que nunca deixou eu me abater, trazendo sempre uma palavra de consolo, por estar sempre comigo e pelas inúmeras noites acordado ao meu lado me ajudando neste trabalho. Obrigada por acreditar em mim.

Aos demais familiares: meu irmão Dener, minha avó Elena e minha tia Berenice pelo apoio em minhas escolhas e por acreditar que tudo daria certo.

A minha orientadora Diana Adamatti que sempre acreditou que eu conseguiria, mesmo quando eu pensava em desistir. Agradeço muito o seu tempo de dedicação, paciência, preocupação comigo e interesse em transmitir sua experiência e conhecimento em prol de meu aprendizado.

E por fim, agradeço a todos que torceram por mim nessa caminhada.

Sumário

Lista de Abreviaturas ou Siglas	ix
Lista de Figuras	x
Lista de Tabelas.....	xi
Resumo	xii
Abstract	xiii
1 Introdução	1
1.1 Justificativa	2
1.2 Objetivos	3
1.3 Organização do texto.....	4
2 Revisão Bibliográfica.....	6
2.1 Oceanografia	6
2.2 Descrição dos Oceanos.....	8
2.3 Oceanos da Terra.....	12
2.4 Modelo OCCAM – Ocean and Circulation Advanced Modeling	15
2.5 Banco de dados	16
2.6 Dicionário de Dados – Modelo de Entidade-Relacionamento	21
3 Metodologia	38
4 Modelo Proposto.....	41
4.1 Dicionário de Dados	41
4.2 Modelo Entidade-Relacionamento e Diagrama Entidade-Relacionamento.....	44
4.3 Projeto Lógico.....	48
4.4 Obtenção dos dados.....	51
4.5 Inserção dos dados no Banco de Dados	61
4.6 Análise dos Dados Armazenados.....	68
5 Conclusão e Trabalhos Futuros.....	71
Referências	73
Anexos	76
Anexo 1. Script das oito tabelas criadas no SGBD PostgreSQL	76
Tabela ITERAÇÃO	76
Tabela COORDENADAS T	77
Tabela COORDENADAS U	78
Tabela DADOS T.....	80
Tabela DADOS U	81
Tabela DADOS COMPLEMENTARES U.....	82
Tabela DADOS COMPLEMENTARES T	84
Tabela PROFUNDIDADE.....	85
Anexo 2. Rotina dos comando que foram utilizados para a visualização dos dados no MATLAB	86

Lista de Abreviaturas ou Siglas

BD	- Banco de Dados
DBA	- Data Base Administrator
ER	- Entidade-Relacionamento
FK	- Foreign Key
FURG	- Universidade Federal de Rio Grande
GFDL	- Linguagem de Descrição de Fluxo de Grade
MB	- Megabytes
MER	- Modelo Entidade-Relacionamento
MOM	- Modular Ocean Model
NetCDF	- Network Common Data Form
NODC	- National Oceanographic Data Center
OCCAM	- Ocean and Circulation Advanced Modeling
PK	- Primary Key
SGBD	- Sistema Gerenciador de Banco de Dados
SQL	- Structured Query Language
XBT	- Expendable Bathythermograph

Lista de Figuras

Figura 2.2 Quantidade relativa ao total de água do planeta, sendo que 97,5 é água salgada dos oceanos e 2,5 água doce.....	13
Figura 2.3 Arquivo de alunos.....	18
Figura 2.5 Representação gráfica de entidades.	23
Figura 2.6 Principais atributos de uma entidade.	24
Figura 2.7 Atributos de uma entidade em forma de diagrama.....	25
Figura 2.10 Atributo derivado.	27
Figura 2.11 Cardinalidade de atributos no diagrama.....	28
Figura 2.12 Representação gráfica de relacionamentos.	29
Figura 2.13 Cardinalidade de relacionamentos.....	29
Figura 2.14 Relacionamento 1:1 (um-para-um).....	30
Figura 2.15 Relacionamento n:n (muitos-para-muitos).	31
Figura 2.16 Cardinalidade de relacionamento ternário.	32
Figura 2.17. Cardinalidade mínima de relacionamento.	33
Figura 2.18 Generalização/especialização.....	34
Figura 2.19 Generalização/especialização total	34
Figura 2.20 Generalização/especialização parcial.....	35
Figura 2.21 Generalização/especialização com herança múltipla.....	35
Figura 2.22 Relações entre tabelas.	36
Figura 3.1 Metodologia dividida em quatro partes.....	38
Figura 4.2 Modelo ER dos dados oceanográficos em representação de tabelas.....	45
Figura 4.3 Acesso com senha.	48
Figura 4.4 Criação de um novo <i>DataBase</i>	49
Figura 4.5 Criação do banco OCCAM.	49
Figura 4.6 Criação das oito tabelas.	50
Figura 4.7 Script da tabela ITERAÇÃO.	50
Figura 4.8 Informações geradas ao declarar o comando <i>ncdisp</i> no MATLAB.	52
Figura 4.9 Informações geradas ao declarar o comando <i>ncdisp</i> no MATLAB.	53
Figura 4.10 Informações geradas ao declarar o comando <i>ncdisp</i> no MATLAB... ..	53
Figura 4.11 Informações geradas ao declarar o comando <i>ncdisp</i> no MATLAB.. ..	54
Figura 4.12 Informações geradas ao declarar o comando <i>ncdisp</i> no MATLAB.. ..	54
Figura 4.13 Informações geradas ao declarar o comando <i>ncdisp</i> no MATLAB.. ..	55
Figura 4.14 Informações geradas ao declarar o comando <i>ncdisp</i> no MATLAB.. ..	55
Figura 4.15 Informações geradas ao declarar o comando <i>ncdisp</i> no MATLAB.. ..	56
Figura 4.16 Informações geradas ao declarar o comando <i>ncdisp</i> no MATLAB.. ..	56
Figura 4.17 Comando <i>ncdisp1</i> no MATLAB.	57
Figura 4.18 Comando para abrir os dados da variável <i>LONGITUDE_T</i>	58
Figura 4.19 Janela do <i>Workspace</i> do MATLAB com os dados armazenados na variável <i>LONGITUDE_T</i>	58
Figura 4.20 Janela do <i>Workspace</i> do MATLAB com os dados armazenados na variável <i>KMT</i> . .	59
Figura 4.21 Janela do <i>Workspace</i> do MATLAB com a continuação dos dados armazenados na variável <i>KMT</i>	59
Figura 4.22 <i>Workspace</i> do MATLAB para visualizar os dados da variável <i>U_VELOCITY_MEAN_</i>	60
Figura 4.24 Janela de comando para a inserção de dados na tabela <i>COORDENADAS_</i> no PostgreSQL.	64
Figura 4.25 Janela do <i>PostgreSQL</i> para a inserção dos dados na tabela <i>COORDENADAS_T</i>	64
Figura 4.26 Tabela preenchida na planilha com os dados a serem inseridos no SQL da tabela <i>COORDENADAS_T</i>	65

Figura 4.27 Janela para visualização dos dados armazenados na tabela COORDENADAS_T.	66
Figura 4.28 Relatório do <i>PostgreSQL</i> do tamanho de armazenamento do banco de dados relacional.....	69
Figura 4.29 Relatório do <i>PostgreSQL</i> do tamanho de armazenamento de cada tabela no banco de dados relacional..	69

Lista de Tabelas

Tabela 2.1 Densidade da água do mar para diferente combinações de temperatura e salinidades Schmiegelow et al. (2004).	10
Tabela 2.4 Dicionário de dados.	22
Tabela 2.8 Atributos de uma entidade em forma de tabela.	25
Tabela 2.9 Hierarquia de atributos compostos.	26
Tabela 4.1 Dicionário de dados do Modelo OCCAM.	41
Tabela 4.23 Tabela com o número identificador para as chaves primárias.....	61

Resumo

Este trabalho tem por objetivo implementar dados oceanográficos em um banco de dados estruturado na forma Modelo Entidade-Relacionamento (MER), o qual permitirá que os dados não se repitam, nem mesmo repliquem uma vez armazenados; e realizar análises comparando o armazenamento atual com a forma estruturada desta técnica (MER). Atualmente, os arquivos se encontram armazenados em um formato *NetCDF* e para cada dia de coleta de dados, há um arquivo, nesse formato, com tamanho de 432 MB de armazenamento. As coletas desses dados foram realizadas com intervalo de 5 dias, do dia primeiro de janeiro de 1988 ao dia 31 de dezembro de 2004, totalizando 1.168 coletas de dados ao longo desses 16 anos. E ainda, em cada coleta há informações de 24 variáveis. Isso significa que o armazenamento destes dados é muito volumoso, e conseqüentemente torna o banco de dados pesado. Para atingir tais objetivos, a metodologia foi dividida em quatro etapas que foram estudos das três áreas do contexto deste trabalho, a modelagem dos dados e do modelo lógico do OCCAM com a implementação da técnica MER no SGBD *PostgreSQL*, a extração dos dados no MATLAB para inserí-los no banco juntamente com análises quanto ao tamanho de armazenamento gerado. Após essas etapas serem alcançadas, concluiu-se que mesmo duas tabelas (com duas variáveis armazenadas em cada uma), não tenham sido incluídas no banco, o tamanho de armazenamento do banco de dados relacional foi de aproximadamente 222 MB, isto é, com aproximadamente 50% do tamanho de armazenamento do original.

Palavras-chave: Dados oceanográficos; Modelo OCCAM; Banco de dados relacional;

Abstract

This work aims to implement oceanographic data in a database structured as Entity-Relationship Model (MER), which will allow data are not repeated, not even replicate once stored; and we perform analysis comparing the current storage and MER structured technique. Currently, the files are stored in a NetCDF format and for each day of data collection, there is a file in this format, with size 432 MB. The collection of such data were made at 5-day intervals, from January, 01 1988 to December, 31 2004, totaling 1,168 data collection over these 16 years. Moreover, these collection there are reports of 24 variables to storage. In this way, this database is too voluminous and heavy. To achieve these objectives, the adopted methodology was divided into four steps that were studies of the three areas of the context of this work, the modeling data and logical model of OCCAM with the implementation of MER technique in PostgreSQL DBMS, data extraction in MATLAB to insert them in the database and the analysis of he generated storage size. Even two tables (with two variables stored in each one), were not included in the database, we could conclude that the storage size of the relational database was approximately 222 MB, i.e. approximately 50% of the original storage size.

Keywords: Oceanographic Data; OCCAM model; Relational Database;

1. INTRODUÇÃO

Desde o início da civilização humana, os oceanos já eram objeto de interesse entre os cientistas, mas o seu estudo só adquiriu forte identidade no Século XX. Seu estudo possui uma grande importância para a humanidade, pois aproximadamente três quartos da Terra são banhados por água, e a grande parte da população do planeta situa-se ao longo da costa. Eles também são fontes de energia com destaque na extração de petróleo e minerais e na rota comercial para o transporte de mercadorias do mundo todo (cerca de 90% do transporte de mercadoria do mundo é feita nos oceanos). Em decorrência disso, surgiu a ciência conhecida como Ciência do Mar: a Oceanografia. O seu estudo é focado nos oceanos e na sua dinâmica, bem como nas inúmeras formas de vida nele existentes. A partir das informações coletadas, procura-se observar, simular e compreender os eventos da natureza para que possam prever seu comportamento e assim gerenciá-lo, caso seja necessário.

Com base nisso, há uma necessidade de obtenção de informações sobre este meio, o que envolve uma grande quantidade de equipamentos e modelos técnicos como numéricos e sensoriamento remoto. Há um modelo numérico do oceano global, denominado Modelo OCCAM (*Ocean and Circulation Advanced Modeling*) desenvolvido por pesquisadores do Centro Nacional de Oceanografia, Southampton, em colaboração com as Universidades de East Anglia e Edimburgo, que vem sendo muito utilizado entre pesquisadores da área de Oceanografia por possibilitar estudos de modelagem numérica em menor escala. Este aninhamento de informações entre modelos numéricos de diferentes escalas vem sendo amplamente utilizado no Laboratório de Oceanografia Costeira e Estuarina da FURG.

Entretanto, estes dados oceanográficos que se originaram no Modelo OCCAM utilizados pelos pesquisadores e estudantes deste laboratório da FURG, dizem respeito a uma região do Oceano Atlântico Sul, localizado no Hemisfério Sul, e que se encontram armazenados em uma grande densidade, por possuir rotinas de coletas (coletas de um intervalo de 5 dias) de dados de 24 variáveis ao longo de 16 anos. Para isso, o objetivo deste trabalho é a implementação desses dados oceanográficos em um banco de dados estruturado na forma do Modelo Entidade-Relacionamento (MER), e realizar análises comparando o armazenamento atual (arquivos compactados no formato *NetCDF*) com a

forma estruturada desta técnica (MER), o que é apresentado e explicado ao longo do trabalho.

A escolha de armazenamento de informações em bancos de dados deve-se ao fato de que os avanços tecnológicos dos últimos anos, possibilitaram que os bancos de dados fossem aplicados de formas inovadoras. Eles possibilitaram aos seus usuários a solicitação para que o sistema realize várias operações que vão desde acrescentar novos arquivos ao banco de dados, como também removê-los do banco; inserir, buscar, excluir e alterar dados de arquivos existentes. E ainda, um banco de dados pode ser criado e mantido utilizando-se um Sistema Gerenciador de Banco de Dados (SGBD).

1.1 Justificativa

O armazenamento de dados em um BD (Banco de Dados) vem sendo muito utilizado nas empresas que substituíram os métodos tradicionais (registro em papéis, arquivos atualizados manualmente) pelo armazenamento em um banco de dados digital. Os BDs proporcionam muitas vantagens aos usuários, e tais vantagens justificam a escolha de um banco de dados relacional para armazenar dados oceanográficos. Dentre elas estão a densidade, atualidade e proteção.

Os dados oceanográficos que serão armazenados no BD se encontram em grande densidade. Isso acontece, porque para cada dia de coleta de dados há um arquivo. E cada arquivo está compactado em um formato *NetCDF* com 432 MB de armazenamento. As coletas desses dados foram realizadas com intervalo de 5 dias, do dia primeiro de janeiro de 1988 a 31 de dezembro de 2004, isto é, a cada ano houve um total de 73 coletas, que ao longo dos 16 anos totalizou 1168 coletas. E ainda, em cada coleta são armazenadas 24 variáveis. Isso significa que o armazenamento para estes dados torna-se muito volumoso, o que torna o BD pesado. Uma solução que poderá diminuir o tamanho de armazenamento é o armazenamento destes dados oceanográficos em um banco de dados relacional. A escolha de utilizar o BD relacional se deve ao fato de que os dados não irão se repetir, nem mesmo replicar, uma vez que já tenham sido armazenados. Isso facilitará a consulta e não exigirá um armazenamento extenso.

Esta modelagem de dados do Modelo OCCAM para a inserção em um BD relacional na forma estruturada do ER será desenvolvida pela primeira vez, tornando este trabalho pioneiro nesse assunto. Há inúmeros trabalhos que optam em utilizar bancos de dados estruturados na forma entidade-relacionamento para armazenar diversos tipos de dados, mas ainda não houve nenhum trabalho que utilizasse desta técnica (MER) para o armazenamento dos dados oceanográficos do Modelo OCCAM. Portanto, os objetivos principais deste trabalho tem o intuito de contribuir no trabalho dos pesquisadores da área de Oceanografia, que utilizam dos dados do Modelo OCCAM, proporcionando um mecanismo de busca das informações de forma mais rápida e organizada, e ainda, sem que demande um grande tamanho de armazenamento dos dados oceanográficos.

1.2 Objetivos

Objetivo Geral

Construir um banco de dados para os dados oceanográficos do Modelo OCCAM com base na técnica de Modelagem Entidade-Relacionamento (MER), realizar análises quantitativas sobre a otimização proporcionada pelo armazenamento dos dados na forma estruturada (MER) em comparação com o método tradicional de armazenamento pelo qual se encontram.

Objetivos Específicos

Os objetivos específicos foram divididos em:

- Entendimento dos principais conceitos oceanográficos e dos dados do modelo OCCAM.
- Elaborar um dicionário de dados especificando todos os dados oceanográficos que serão armazenados no banco de dados relacional.
- Construir um modelo conceitual, na forma de diagrama entidade-relacionamento (diagrama ER) com os dados obtidos.
- Transformar o modelo conceitual em um modelo lógico para implementá-lo em um Sistema Gerenciador de Banco de Dados (SGBD). Realizar análises

comparando a otimização obtida com a estruturação do banco de dados relacional e o método atual de armazenamento de dados.

1.3 Organização do Texto

A Dissertação está organizada em capítulos de revisão bibliográfica, metodologia, modelo proposto, conclusão e trabalhos futuros. O Capítulo 2, de revisão bibliográfica, está subdividido em três grandes áreas de estudo, as quais são: oceanos, bancos de dados, e modelo OCCAM. Para a primeira área, é apresentada uma revisão sobre a Oceanografia e os seus campos de estudos; em seguida, aborda-se a origem, história e descrição dos oceanos, com seus principais parâmetros termodinâmicos: salinidade, temperatura, densidade, pressão e tensão dos ventos sobre o oceano. E, além disso, é apresentada rápida explanação do hemisfério sul, foco dos dados coletados nesse projeto. A segunda área de estudo abrange o embasamento sobre o Modelo OCCAM, e a referência de trabalhos que utilizaram este modelo em suas pesquisas. A terceira área envolve banco de dados. Para isso, houve a necessidade de apresentar os conceitos básicos de banco de dados, o Modelo de Entidade Relacionamento, o dicionário de dados e o diagrama de entidade-relacionamento, sendo que estas são etapas essenciais para a compreensão de um banco de dados relacional.

No Capítulo 3 será apresentada a metodologia utilizada para este trabalho, representada em um fluxograma, que consiste em um processo de quatro etapas crescentes. A primeira e a segunda etapas são respectivamente, o estudo das três áreas que envolvem este trabalho (presente no capítulo da revisão bibliográfica) e a modelagem dos dados, isto é, a criação do dicionário de dados e do modelo de entidade relacionamento. A terceira etapa consistiu em converter as informações dos dados oceanográficos para a base de dados relacional, onde optou-se pelo SGBD *PostgreSQL*. A etapa final consistiu na análise dos resultados obtidos ao longo da implementação do SGBD, em comparação com o método atual.

O Capítulo 4, Modelo Proposto, apresenta o dicionário de dados, e o Modelo de Entidade Relacionamento, e os scripts utilizados no SGBD (passo-a-passo) para a criação do BD relacional. E ainda se encontram neste capítulo a obtenção e visualização dos dados através do toolbox do MATLAB, juntamente com alguns comandos

utilizados nele e a explicação de como foi a exportação dos dados do Modelo para o banco de dados do *PostgreSQL*.

No Capítulo 5, são apresentados os resultados alcançados no trabalho, apresentando a comparação do tamanho de armazenamento dos dados em formato NetCDF e os do banco de dados relacional. E, por fim, no último capítulo estão as conclusões deste trabalho, juntamente com a discussão dos trabalhos futuros.

2. REVISÃO BIBLIOGRÁFICA

Este capítulo aborda assuntos relevantes nas três áreas de estudo que são abordadas no contexto do projeto. Desta forma, será apresentada uma visão geral sobre alguns dos principais aspectos oceanográficos relacionados ao estudo proposto, com destaque para a importância da oceanografia e as suas grandes áreas de estudo, bem como para a apresentação da origem, história e descrição dos oceanos. Especial enfoque será dado ao comportamento dos oceanos no Hemisfério Sul. Na sequência, tem-se a apresentação do modelo OCCAM (*Ocean and Circulation Advanced Modeling*), do qual foram extraídos os dados utilizados neste trabalho, e ainda são citados alguns trabalhos baseados nos dados deste Modelo. Por fim, é apresentada uma revisão sobre banco de dados, seus conceitos básicos necessários para a compreensão ao longo do trabalho, a abordagem e compreensão do Modelo Entidade-Relacionamento, do dicionário de dados e do diagrama entidade-relacionamento.

2.1 Oceanografia

A Oceanografia também pode ser chamada de Ciência do Mar, atuando no estudo dos oceanos e da sua dinâmica, bem como das inúmeras formas de vida nele existentes. Garrison (2010) define Oceanografia como sendo “o processo de descoberta de princípios unificadores nos dados obtidos em pesquisas no oceano, as formas de vidas existentes e as áreas de Terra que o limitam”. A partir dessas informações, procura-se observar, simular e compreender os eventos da natureza para que possam prever seu comportamento e assim gerenciá-lo, caso seja necessário.

Ao longo do tempo, desde o início da civilização humana, os oceanos já eram objeto de interesse entre os cientistas, mas o seu estudo só adquiriu forte identidade no Século XX. O estudo desta ciência é de suma importância para a humanidade, pois aproximadamente três quartos da Terra são banhados por água, e a grande parte da população do planeta situa-se ao longo da costa. É também dos oceanos que são retiradas inúmeras fontes de energia, tais como a extração de petróleo e minerais, e eles também servem de rota comercial para mais de 90% da frota de mercadorias no mundo. Atualmente, grande destaque é dado ao efeito das mudanças climáticas (ciclos de El

Niño e La Niña, furacões, tempestades tropicais, entre outros), sobre os oceanos e suas consequências para a humanidade (Martins, 2009).

A oceanografia é considerada uma área multidisciplinar por permitir o envolvimento de objetos de estudos de outras disciplinas, tais como a Física, Química, Geologia, Biologia e a Engenharia. Schmiegelow (2004) enfatiza que “a Oceanografia é a aplicação de todas as ciências para entender o fenômeno dos oceanos; é considerada uma ciência multidisciplinar.” A mesma está dividida em quatro grandes áreas de estudo: Oceanografia Física, Oceanografia Química, Oceanografia Biológica e Oceanografia Geológica, e também a Geofísica. As mesmas serão apresentadas, respectivamente.

A Oceanografia Física concentra-se no estudo e observação do movimento das águas, ou seja, as correntes, as ondas e as marés, e também as interações entre a atmosfera e oceano, a propagação do som e da luz na água. Schmiegelow (2004) destaca que esta é a área da Oceanografia onde encontra-se o maior desenvolvimento tecnológico. Frequentemente, e cada vez mais, analisa-se em escala global, os movimentos da água, necessitando-se por isso, da análise de grandes quantidades de dados em pouco tempo, já que tais fenômenos são bastante dinâmicos. Além disso, em diversos estudos do ambiente, utiliza-se de equipamentos eletrônicos muito sensíveis e de alta tecnologia, como sistemas de observação in situ, boias oceanográficas e dados de sensoriamento remoto (Portal, 2009).

A Oceanografia Química estuda a relação entre os aspectos biológicos e geológicos do oceano, e também identifica os componentes e elementos presentes na água do mar para análises mais precisas de suas concentrações, auxiliando no entendimento do ambiente químico de mar ao ser afetado pela ação do homem, sendo importante na determinação de poluentes, como o mercúrio, materiais radioativos e o petróleo.

A Oceanografia Biológica é considerada a área de trabalho mais complexa dentro da Oceanografia, pois abrange todas as formas de vida dos oceanos e o estudo da sua dinâmica. Mafalda (2012) comenta que a Oceanografia Biológica tem o papel de descrever a distribuição, abundância e produção das populações e compreender os processos que as influenciam. O seu estudo está centralizado na pesquisa dos organismos marinhos, a sua distribuição e ciclos de vida. Ela também se preocupa em

estudar os impactos a esses organismos causados por poluentes do oceano e da atmosfera, pela indústria pesqueira e outros.

A Oceanografia Geológica se interessa principalmente pelo estudo das rochas e sedimentos marinhos, o qual pode permitir a compreensão da formação dos oceanos e a composição do interior de nosso planeta. Segundo Schmiegelow (2004), “a Oceanografia Geológica estuda também o substrato oceânico com relação aos aspectos profundos da crosta (estrutura e propriedades físicas através de características magnéticas e gravitacionais).” E para isso, utiliza-se de ferramentas acústicas muito sofisticadas. No estudo desta área também se faz necessário a interação com a Geofísica e a Geoquímica.

2.2 Descrição dos oceanos

Acredita-se que desde a civilização humana, os oceanos eram objeto de estudo. Entretanto, foi nas últimas décadas que houveram estudos mais abrangentes que levaram a descobertas de inúmeros fatores com grandes relações em fenômenos naturais, em que muitas das vezes, eram dadas as causas desses fenômenos por desconhecidas. Com isso, possibilitou uma visão ainda mais ampla de como os oceanos afetam a atmosfera.

Existe uma grande interação entre a atmosfera e os oceanos. Schmiegelow (2004) comenta que os raios solares aquecem a atmosfera, os solos e os oceanos uma e meia a duas vezes mais por unidade de área nas regiões equatoriais do que nas polares. Isto gera um balanço energético responsável por transferir o calor que é recebido pelo equador para os polos. Este mecanismo é um dos principais responsáveis pelo surgimento das grandes correntes oceânicas do planeta e também dos principais cinturões de vento: os ventos alísios sopram do leste para o oeste e ocorrem entre uma latitude de 0° e 30°; os ventos do Oeste que sopram do oeste para o leste com uma latitude entre 30° a 60°; e por fim os ventos do leste, os quais sopram do leste para o oeste nas regiões polares. Portanto é a atmosfera a principal responsável pelo transporte de calor das regiões do Equador para as regiões polares.

A Salinidade é a medida de quantidade de sal presente nas águas do oceano, sendo de grande importância à oceanografia, pois ela é essencial à identificação das massas da

água do mar e no estudo do seu comportamento e mistura, por exemplo. Garrison (2010) explica que a salinidade dos oceanos varia de cerca 33 a 37 PSU (Escala Prática de Salinidade - PSU), dependendo de fatores como evaporação, precipitação e descarga de água doce dos continentes. Entretanto, a salinidade média está geralmente ao redor de 35 PSU. Isso significa que, se retirássemos toda a água que compõe o mar, iria se formar uma enorme camada de sais espalhados nas áreas oceânicas, cobrindo cerca de 75% da superfície do nosso planeta. Em regiões costeiras, onde ocorrem descargas de águas doces dos rios, a salinidade é mais baixa, podendo chegar a 20 PSU.

A temperatura da água dos oceanos varia tanto horizontalmente quanto verticalmente na coluna d'água. Juntamente com a salinidade, a temperatura é determinante para a densidade da água do mar, o que se percebe quando Schmiegelow (2004) exemplifica que “a salinidade afeta a temperatura na qual a água do mar se congela”. Isso explica porque ambas não podem ser abordadas separadamente. E ainda, o autor ressalta que a salinidade e a temperatura podem ser úteis para identificação das massas de água e dos sistemas de correntes.

Para se entender melhor a distribuição de temperatura da água do mar, deve-se entender como funciona a Distribuição Horizontal e Vertical, onde ambas são tratadas de formas diferentes. Na Distribuição Horizontal, a superfície oceânica é aquecida pelo Sol e pelo calor da atmosfera. Contudo, ao mesmo tempo em que a superfície recebe calor, o contrário também ocorre com o resfriamento superficial em decorrência da radiação de retorno da superfície com atmosfera.

A temperatura dos oceanos é em média muito baixa, com valores que podem atingir temperaturas menores que 4°C. Outro importante fator, que influencia na temperatura da superfície, é a latitude, pois o Equador, por possuir um ângulo de incidência de radiação maior em pequenas latitudes do que nas regiões polares, faz com que ele receba mais calor por unidade de área. Portanto, quanto mais próximo da região equatorial, maior será a temperatura superficial da água.

O aquecimento superficial do oceano se divide em três camadas na coluna d'água: a camada isotérmica, a termoclina principal e a camada de fundo; e ainda poderá haver uma outra camada de mistura nas latitudes pequenas e médias, que é a termoclina sazonal. A camada isotérmica é formada a partir da rápida transferência de calor nas

regiões da superfície, sendo assim a temperatura se mantém igual nessa região. A mesma é denominada de camada de mistura localizada a aproximadamente a 100 metros de profundidade. Na termoclina superficial, a temperatura cai inesperadamente com a profundidade, e ela se encontra entre a superfície aquecida pelo Sol e as profundas águas frias do oceano. A região que atinge o fundo do oceano é denominada camada do fundo, onde a temperatura varia entre 1°C e 3°C em profundidades que possam passar dos 4.000 metros. Por fim, a camada de mistura, que também pode ser originada em decorrência do aumento de temperatura das águas da superfície oceânica em meses que são mais quentes do ano, é a termoclina sazonal.

Minster (1997) apresenta uma comparação entre os oceanos quanto à diferença de temperatura e salinidade, destacando que devido às suas águas do Pacífico são menos densas do que as do Atlântico, que por sua vez, são menos densas do que as do Mediterrâneo.

Atualmente, uma das simples formas mais utilizadas para medir a temperatura dos oceanos é o aparelho conhecido por XBT, uma sonda lançada do navio que mede o perfil de temperatura ao longo da coluna d'água, sendo descartado no mar automaticamente, e também em técnicas de sensoriamento remoto.

Dentre as três propriedades físicas mais importantes do oceano, estão a temperatura, a salinidade e a densidade. Conforme já apresentado anteriormente, a temperatura e salinidade são responsáveis por determinar a densidade, segundo a Tabela 2.1.

Tabela 2.1: Densidade da água do mar para diferentes combinações de temperaturas e salinidades Schmiegelow et al. (2004)

TEMPERATURA (°C)							
SALINIDADE	0	5	10	15	20	25	30
5	1,00397	1,00401	1,00367	1,00301	1,00207	1,00087	0,99943
10	1,00801	1,00796	1,00756	1,00684	1,00585	1,00461	1,00314

15	1,01204	1,01191	1,01144	1,01067	1,00964	1,00836	1,00686
20	1,01606	1,01585	1,01531	1,01449	1,01341	1,01210	1,01057
25	1,02008	1,01980	1,01920	1,01832	1,01720	1,01585	1,01429
30	1,02410	1,02374	1,02308	1,02215	1,02099	1,01960	1,01801
35	1,02813	1,02770	1,02697	1,02600	1,02479	1,02337	1,02176
40	1,03216	1,03166	1,03088	1,02985	1,02860	1,02715	1,02551

Observando a tabela 2.1, tem-se que a densidade da água pura cresce de acordo com a diminuição da temperatura. Contudo Schmiegelow (2004) explica que, para a água pura, a densidade aumenta com a diminuição de temperatura até o valor máximo de 4 °C, quando então decresce em direção ao ponto de congelamento a 0 °C. Para a água salgada, entretanto, a densidade continua a aumentar em direção ao ponto de congelamento. Devido aos sais dissolvidos, o ponto de congelamento se reduz para cerca de -1,9 °C (para a água do mar com salinidade de 35 PSU). Embora com menor importância, a pressão é também uma variável que influencia a densidade. Garrison (2010) acrescenta que a densidade da água do mar aumenta com a elevação de salinidade, o aumento da pressão e o decréscimo da temperatura. Sendo assim, existem diversos fatores que podem influenciar a densidade. Entre eles estão a descarga de água doce dos rios nos oceanos, processos de evaporação e aquecimento, dissolução de gelo e pressão.

A pressão pode ser compreendida como sendo uma função do peso da coluna d'água, e conseqüentemente, também da profundidade e da densidade da água do mar. A cada 10 metros de profundidade, a pressão irá aumentar 1atm (atmosfera). Skinner e Turekian (1977) explicam que a densidade da água do mar é de cerca de 1,03 g/cm³ (é 3% mais alta do que a água doce devido ao sal dissolvido), de modo que cada intervalo de 10 m de profundidade da água do mar iguala-se à massa da atmosfera, e causará um aumento de pressão igual à pressão experimentada na superfície. A massa atmosférica, sob a ação do campo gravitacional terrestre exerce uma força. A força por área unitária é

chamada de *pressão*. A pressão média da atmosfera terrestre ao nível do mar é chamada uma “atmosfera”, de modo que a 10 m de profundidade a pressão sentida por um mergulhador seria de duas “atmosferas”. A 500 m de profundidade a pressão será, então, de 51 atm. Esta unidade *atm* utilizada na medida da pressão decorre a partir da pressão exercida através de uma coluna de mercúrio de 76 centímetros de altura de uma área de 1 cm^2 para uma temperatura de 0°C .

Os seres que habitam nas profundezas dos oceanos estão aptos a suportar pressões que podem ultrapassar 1100 *atm*, o que significa um peso de 1,1 toneladas por ton/cm^2 centímetros quadrados. Já os seres vivos terrestres, praticamente nunca foram submetidos a pressões superiores a 1 *atm*.

Em relação ao vento agindo na superfície dos oceanos, a sua velocidade cresce de acordo com a altitude em baixas camadas da atmosfera, e esta tensão do vento transfere energia para a coluna de água, gerando ondas e correntes. Minster (1997) explica que a velocidade do vento cresce com a altitude nas baixas camadas da atmosfera, justamente porque o ar afrouxa à superfície por força do atrito: escolhe-se arbitrariamente a velocidade do vento a 10 m de altitude para exprimir o seu valor. Esta tensão equivale a $0,1\text{ N/m}^2$ (Newton por metro quadrado), considerando uma velocidade de vento de 8 m/s (metros por segundo).

2.3 Oceanos da Terra

Acredita-se que há 4600 bilhões de anos, período após a formação da Terra, enormes quantidades de vapor de água foram liberadas pelos vulcões e pela superfície do planeta, que se encontrava em fogo. Após passarem milhões de anos, ocorreu uma condensação do vapor por consequência da temperatura da superfície da Terra ter baixado. Sendo esta a explicação da existência de toda água que se tem hoje, o qual foi chamado de “oceano primitivo”, onde era quase impossível que houvesse um sinal de vida, pois suas águas eram ácidas com temperaturas a ponto de ebulição, totalmente diferente dos oceanos atuais. Desde então, a atmosfera terrestre não parou de promover alterações nas suas características. O gás carbônico se diluiu na água, onde atualmente se faz presente em maior quantidade. De todo o gás carbônico que está na Terra, aproximadamente 99% dele está concentrado nas águas do oceano.

O nosso planeta deveria ser chamado de “planeta água”, pois toda a água dos oceanos ocupa quase 3/4 da superfície da Terra, isto é, cerca de 70,8% da superfície de nosso planeta é coberta por água. De toda a água presente na superfície terrestre, 97% pertence somente aos oceanos, com um volume de 1.370.000.000 Km³. A outra parte da água de nosso planeta está distribuída entre rios e lagos, que correspondem a 0,007%, 1,7% em forma de gelo, 0,8% na água subterrânea e 0,001% encontrada na atmosfera. Entretanto, 97,5% de toda água é salgada, e apenas 2,5% corresponde a água doce, conforme mostra a figura 2.2.

Total de Água na Terra

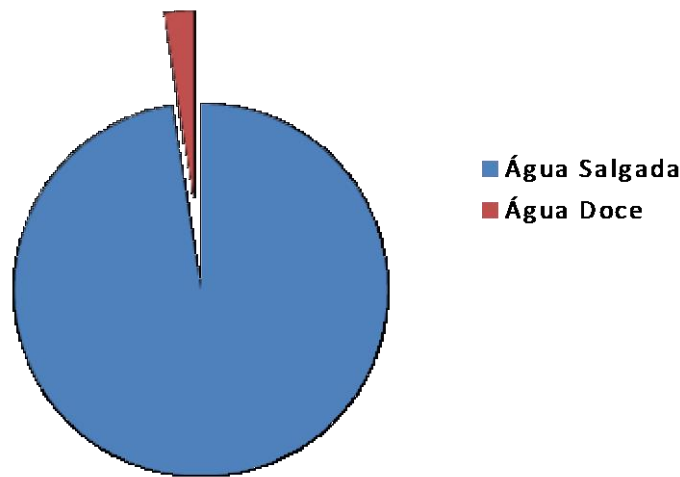


Figura 2.2 Quantidade relativa ao total de água do planeta, sendo que 97,5 é água salgada dos oceanos e 2,5 água doce.

Garrison (2010) destaca que na escala humana, o oceano é incrivelmente grande, pois cobre 361 milhões de quilômetros quadrados da superfície da Terra. A profundidade média é de 3.796 m, o volume de água é de 1,37 bilhões de Km³, e a temperatura média é de 3,9 °C. A massa corresponde 141 bilhões de toneladas. Se a superfície da Terra fosse nivelada, como uma bola, o oceano a cobriria com uma profundidade de 2.686 metros. A elevação média da Terra é de apenas 840 metros, mas a média da profundidade do oceano é de 4,5 vezes maior.

Dentre os oceanos do nosso planeta, estão o Oceano Atlântico e o Oceano Índico, conhecidos desde a Antiguidade, e Oceano Pacífico, descoberto no século XVI, e os Oceanos que circulam os pólos: o Glacial Ártico, e o Antártico.

O Oceano Atlântico é considerado o segundo maior oceano em extensão do mundo, com seus 80 milhões de Km², estando localizado entre a África, América (oeste) e Europa (Leste). Seu nome é de origem grega, deriva do nome Atlas, que era um titã da mitologia grega, sendo por este motivo chamado de “Mar de Atlas”.

O Atlântico banha aproximadamente 20% da superfície global, tornando-se o mais importante por causa de sua localização, por ter 70% das rotas marítimas comerciais do mundo, por possuir uma rica biodiversidade de espécies marinhas, e ainda por possibilitar a pesca e o turismo. Outra característica importante deste oceano é que vários rios desembocam nele, totalizando uma média de 600 mil m³ por segundo. Dentre estes rios, está o rio Amazonas, responsável por desaguar ¼ do total de água doce vinda dos rios. Este rio, juntamente com o rio Congo da África, possuem juntos ¼ de toda água doce da Terra. Como seus vizinhos, incluem o mar do Caribe, o mar do Báltico, o mar do Norte e o Mediterrâneo.

O Oceano Índico está situado principalmente no Hemisfério Sul, e tem uma extensão de 73.440.000 Km², com fronteiras que vão desde o sul da África do Sul até os limites com o Oceano Pacífico. Segundo Schmiegelow (2004), este oceano, como o Atlântico, contém poucas ilhas e é marcado por numerosos platôs submarinos e grandes sopés continentais. Sua fossa mais profunda é a de Java, com 7450 metros.

Este oceano possui características oceanográficas particulares, tais como em virtude da sua temperatura se manter sempre acima dos 20°C ao longo do ano todo, o que propicia a existência de recifes e ilhas de coral. Contudo, ainda convém destacar que o Índico possibilita o transporte de inúmeras mercadorias, como uma das mais valiosas, o petróleo do sudeste asiático aos países do Ocidente.

O Oceano Pacífico banha cerca de 42% da superfície da Terra, onde nele se encontram as maiores profundidades do Planeta, as quais são chamadas de fossas. Uma delas, conhecida como fossa das Marianas, localizada ao largo das ilhas Marianas, no centro do Pacífico, possui uma profundidade de 11.033 metros. O Pacífico destaca-se por ter regiões de distúrbios sísmicos, isto é, possui regiões muito propícias a abalos causados por deslocamentos das placas tectônicas na superfície terrestre.

Este oceano é considerado o maior dos oceanos por sua área abranger mais de 1/3 da superfície da Terra, mais especificamente, possui uma área de 181.000.000 Km², com um volume de 712.000.000 Km³. Outra de suas características importantes é o seu perímetro, conhecido como cinturão de fogo, pois é a região mais ativa do planeta.

Os dados oceanográficos que serão utilizados neste trabalho dizem respeito a uma região do Oceano Atlântico Sul, localizado no Hemisfério Sul que se originam no Modelo OCCAM.

2.4 Modelo OCCAM – Ocean and Circulation Advanced Modeling

O Modelo OCCAM (*Ocean and Circulation Advanced Modeling*), é um modelo numérico do oceano global desenvolvido por pesquisadores do Centro Nacional de Oceanografia, Southampton, em colaboração com as Universidades de East Anglia e Edimburgo. Segundo Cirano et al. (2006), este modelo possibilita a realização de estudos de modelagem numérica em menor escala, pois fornece condições iniciais e de contorno para outros modelos numéricos. Este aninhamento de informações entre modelos numéricos de diferentes escalas vem sendo amplamente utilizado no Laboratório de Oceanografia Costeira e Estuarina da FURG.

O Modelo OCCAM baseia-se na versão GFDL (Linguagem de Descrição de Fluxo de Grade) do modelo oceânico MOM (Modular Ocean Model), mas inclui uma superfície livre e melhorada dos sistemas de advecção. Uma grade de longitude-latitude regular é usada para o Pacífico, Índico e Atlântico Sul. Uma grade de longitude-latitude rodada é usada para o Ártico e Oceano Atlântico Norte.

O modelo possibilita uma resolução horizontal equivalente a 0,0833° (1° equivale a 111 Km) e uma resolução vertical de 66 níveis ao longo da profundidade. A resolução temporal dos dados é de cinco em cinco dias, ao longo de 16 anos (01/01/1988 a 31/12/2004), com as seguintes variáveis de saída: nível, tensão do vento, balanço de massa d'água, profundidade da camada de mistura (máximo, mínima e média), temperatura, salinidade e correntes U e V.

O Modelo OCCAM vêm sendo utilizado por inúmeros pesquisadores, para o estudo de regiões oceânicas e costeiras, conforme Cirano et al. (2006) afirma:

O OCCAM (*Ocean Circulation and Climate Advanced Modelling Project*) é um dos modelos de circulação global que vem sendo bastante utilizado pela comunidade oceanográfica brasileira, principalmente em estudos de modelagem numérica regional, onde fornece condições iniciais e de contorno para modelos numéricos mais detalhados da circulação. (CIRANO et al., 2006, p.1)

Akhir (2012) utilizou os dados do Modelo OCCAM com o objetivo de ilustrar o padrão sazonal da corrente e temperatura superficial no mar do sul da China. Chapman et al. (1985), mostraram como os dados do Modelo OCCAM podem ser utilizados para auxiliar no desenvolvimento de algoritmos de processamento de sinal que conduzem a uma redução substancial do tempo computacional.

O trabalho de Cirano et al. (2006) teve como objetivos comparar em escala anual os dados do OCCAM com: i) os dados climatológicos de temperatura e salinidade do NODC (*National Oceanographic Data Center*) e ii) os transportes de volume associados a estas massas de água, de acordo com os valores disponíveis na literatura. Os níveis de comparação foram escolhidos de forma a representar o núcleo das principais massas de água que compõem o Atlântico Sul e os sistemas de correntes que são descritos através dos transportes de volume associados às massas de água. Os principais resultados indicam que o modelo é capaz de representar realisticamente a estrutura vertical das principais correntes na região de estudo e as massas de água associadas. E ainda, Drijfhout et al. (2005) realizaram um estudo com base nos dados do modelo OCCAM sobre a origem das águas intermédias que cruzam o Atlântico Equatorial.

2.5 Banco de Dados

Os bancos de dados e os sistemas de banco de dados são componentes essenciais em nossa sociedade cada vez mais tecnológica, conforme Elmasri et al. (2005) destaca. As pessoas, por inúmeras vezes ao dia, se deparam com situações que envolvem banco de dados sem ao menos perceber isso. Ao fazer um depósito no banco ou retirar dinheiro estaremos nos envolvendo com uma interação com os bancos de dados, já que atividades como estas envolvem uma pessoa ou um programa de computador que o qual

acessará um determinado banco de dados. E ainda, até mesmo ao fazer comprar em um supermercado, se esta envolvendo com o contexto de banco de dados, já que na maioria das vezes ao efetuar o pagamento, haverá sempre uma atualização eletrônica dos produtos para se ter um controle de estoque.

Por situações como estas, Elmasri et al. (2005) denominam exemplos como estes do dia-a-dia das pessoas como sendo aplicações tradicionais de banco de dados, onde a maioria das informações, as quais são armazenadas, e acessadas são apresentadas em formato de texto ou em formato numérico. Os autores ainda enfatizam que as aplicações inovadoras e interessantes dos sistemas de banco de dados só são possíveis devido aos avanços tecnológicos dos últimos anos.

Os bancos de dados de multimídia podem, agora, armazenar figuras, vídeos e mensagens sonoras. Os sistemas de informações geográficas são capazes de armazenar e analisar mapas, dados do tempo e imagens de satélite. Os data warehouses e os online analytical processing (OLAP) - processamento analítico on-line - são utilizados em muitas empresas para extrair e analisar as informações úteis dos bancos de dados para a tomada de decisões. (Elmasri et al., 2005, p. 3)

Sendo assim, Date (2003) define que sistema de banco de dados são um sistema computadorizado de manutenção de registros. Já o banco de dados, é considerado por ele como sendo um equivalente eletrônico de um arquivo, isto é, uma coleção de arquivos de dados armazenados em arquivos digitais, onde seus usuários podem solicitar que o sistema realize várias operações, que vão desde acrescentar novos arquivos ao banco de dados como também removê-los do banco; inserir, buscar, excluir e alterar dados de arquivos existentes.

E ainda, Elmasri et al. (2005) defendem que "Um banco de dados é projetado, construído e povoado por dados, atendendo a uma proposta específica. Possui um grupo de usuários definido e algumas aplicações preconcebidas, de acordo com o interesse desse grupo de usuários".

Um banco de dados pode ser criado e mantido tanto manualmente como pode ser computadorizado. Este último permite ser criado e mantido através de aplicativos desenvolvidos especialmente a esse tipo de serviço ou a um sistema gerenciador de banco de dados (SGBD).

Os termos “dados” e “informações” podem ser tratados como palavras sinônimas neste contexto de banco de dados, já que ambas podem ser consideradas qualquer coisa que venha a auxiliar ou ter algum tipo de significado ao usuário ou a organização do sistema a qual servir. Mas há os que preferem distinguí-las, tratando o termo “dados” como sendo o que é realmente armazenado no banco de dados e o termo “informações” como o significado dos dados que o usuário determina. No entanto, optou-se ao longo deste trabalho utilizar ambos os termos como sinônimos.

Para a criação de um sistema de banco de dados deve-se considerar quatro componentes principais, os quais são: dados, hardware, software e usuários. Esses quatro componentes serão apresentados a seguir:

Dados

Os dados armazenados em um banco de dados, normalmente estarão integrados e compartilhados. Date (2003) denomina a seguir cada um deles.

Por integrado, queremos dizer que o banco de dados pode ser considerado como uma unificação de vários arquivos que, de outro modo, serão distintos, com a eliminação de qualquer redundância parcial ou total entre esses arquivos. (Date, 2003, p. 7)

Isso significa que, por exemplo, se há em um banco de dados de uma escola um arquivo chamado ALUNO, que fornece o nome, idade, filiação, endereço, série e outros itens a respeito da pessoa do aluno, e ainda outro arquivo de nome MATRÍCULA, o qual fornece o nome, notas e disciplinas, (ver Figura 2.3); ao realizar uma consulta para saber em qual série um determinado aluno está a partir de suas notas, não precisará incluir essa informação, que será algo repetitivo no arquivo MATRÍCULA, pois uma vez que as informações já foram armazenadas no arquivo ALUNO, não há necessidade de repetí-las, pois estas serão sempre obtidas através deste mesmo arquivo.

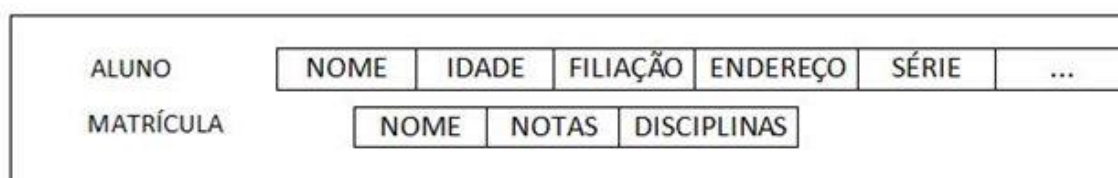


Figura 2.3: Arquivo de alunos

Date explica como funciona o compartilhamento de banco de dados:

Por compartilhado, queremos dizer que o banco de dados pode ser compartilhado entre diferentes usuários, no sentido de que diferentes usuários podem ter acesso aos mesmos dados, possivelmente ao mesmo tempo ("acesso concorrente"). Tal compartilhamento, concorrente ou não, em parte é uma consequência do fato de que o banco de dados é integrado. (Date, 2003, p. 7)

Hardware

O hardware nada mais é do que a parte física do computador. Ele possui vários tipos com diferentes tipos de funcionalidades e objetivos. Para este caso, os componentes de hardware do sistema, segundo Date (2003), consistem em:

Volumes de armazenamento secundário – normalmente, discos magnéticos -, que são usados para manter os dados armazenados, juntamente com os dispositivos de E/S (entrada/saída) associados (unidades de disco etc.), controlados de dispositivos, canais de E/S e assim por diante.

Processador(es) de hardware e memória principal associada, que são usados para dar suporte à execução do software do sistema de banco de dados. (Date, 2003, p.8)

Software

Para criar e manter um banco de dados existe um determinado software, isto é, um sistema gerenciador de banco de dados (SGBD). Este, segundo Elmasri et al. (2005), é uma coleção de programas, os quais facilitam todos os processos de definição, construção, manipulação e compartilhamento de um banco de dados entre inúmeros usuários.

Há inúmeros outros componentes de software que possuem ferramentas de desenvolvimento de aplicações, recursos que auxiliam em projetos e vários utilitários, mas entre todos eles, o mais importante é o SGBD. Este sistema de software torna-se essencial na criação de um banco de dados, pois ele facilita desde os processos de definição, construção, manipulação e compartilhamento como também possui funções de proteção e manutenção do usuário para com o banco.

Usuários

Para a criação, uso e manutenção de um banco de dados envolvem-se inúmeras pessoas cada uma com suas devidas funções no projeto. Para isso, a cada tipo de função há uma classificação de um determinado grupo de usuário, sendo um total de três classes gerais de usuários.

Date (2003) define a primeira classe como sendo a classe de programadores de aplicação, que é responsável pela escrita de programas de aplicações dos bancos de dados com algum tipo de linguagem de programação.

Primeiro, há os **programadores de aplicações**, responsáveis pela escrita de programas de aplicações de banco de dados em alguma linguagem de programação, como COBOL, PL/I, C ++, Java ou alguma linguagem de alto nível de “quarta geração”. Esses programas acessam o banco de dados emitindo a requisição apropriada (normalmente, uma instrução SQL) ao SGBD. Os programas propriamente ditos podem ser aplicações convencionais em batch, ou então podem ser aplicações **on-line**, cuja finalidade é permitir que um usuário final acesse o banco de dados interativamente (por exemplo, a partir de uma estação de trabalho ou terminal on-line, ou de micro-computador). A maioria das aplicações modernas é do tipo on-line. (Date, 2003, p.9)

Em seguida, vem a classe dos usuários finais, que são responsáveis pelo acesso ao banco de dados, com enfoque nas consultas, atualizações e relatórios. Segundo o autor, há várias categorias de usuários finais, os quais são descritos em uma classificação de usuários finais causais, usuários finais parametrizáveis, usuários finais sofisticados e usuários autônomos.

Os usuários finais causais acionam o banco de dados ocasionalmente, mas precisam de informações diferentes a cada acesso. Eles usam uma linguagem de consulta a banco de dados sofisticada para especificar suas solicitações, e normalmente são gerentes de nível médio ou elevado ou outros profissionais com necessidades ocasionais. (Elmasri et al., 2005, p. 10)

Os usuários finais parametrizáveis são aqueles responsáveis pela consulta e atualização das informações do banco de dados. Estas consultas e atualizações são todas padronizadas, as quais já foram todas testadas e programadas cuidadosamente para que nada possa sair errado na consulta.

Os usuários finais sofisticados são compostos por um grupo de profissionais como engenheiros, cientistas e analistas de negócios, responsáveis por implementar

aplicações que atendam as solicitações complexas .

Os usuários autônomos mantêm um banco de dados pessoal por meio do uso de pacotes de programas prontos que possuem interfaces gráficas ou programas baseados em menus fáceis de usar. (Elmasri et al., 2005, p. 11)

Por fim, a última classe de usuários denominada administrador de banco de dados, ou DBA (de *Data Base Administrator*), é responsável por autorizar, coordenar e monitorar o acesso ao banco de dados e ainda por adquirir recursos de software e hardware quando necessário. Ao administrador também compete à responsabilidade de problemas que envolvem falhas na segurança ou por algum tipo de demora de resposta no sistema.

2.6 Dicionário de dados – Modelo de Entidade-Relacionamento

A técnica mais utilizada e difundida, segundo Heuser (1999), de modelagem de dados para banco de dados é a abordagem de Entidade-Relacionamento (ER). Ela consiste em um modelo de dados representado através de um modelo de entidade-relacionamento (modelo ER).

Usualmente, um modelo ER é representado graficamente, através de um Diagrama Entidade-Relacionamento (DER). A abordagem ER foi criada em 1976 por Peter Chen. Ela pode ser considerada como um padrão de fato para modelagem conceitual. Mesmo as técnicas de modelagem orientadas a objetos, que têm surgido nos últimos anos, baseiam-se nos conceitos da abordagem ER. (Heuser, 1999, p. 11)

Para a modelagem de dados, em banco de dados, há a necessidade desta técnica de modelagem de Entidade e Relacionamento, juntamente com um modelo de explicação de todos os objetos (dados) que pertencem ao banco de dados em questão. Este documento, denominado dicionário de dados, permite ao analista obter uma informação clara sobre todos os objetos do modelo em uma forma textual. Muitas vezes, há algum tipo de objeto presente no modelo que é difícil de entender graficamente, então o dicionário serve para facilitar o entendimento, como é o caso deste trabalho.

O dicionário de dados deve conter, na maioria das vezes, uma lista organizada em ordem alfabética de cada objeto que faz parte do banco de dados, seguido de algum tipo de informação tal como descrição, índices e alocações de espaço. Para a elaboração

do modelo de entidade relacional, tem-se como exemplo um banco de dados de uma escola, onde utilizou-se o dicionário de todos os objetos que estão presentes em um banco de dados de uma escola que neste caso, estão separados por tabelas que se relacionam e não em um dicionário com os objetos definidos listados organizadamente em ordem alfabética. Para este caso, ambos são corretos (ver Tabela 2.4).

Tabela 2.4: Dicionário de dados

DICIONÁRIO DE DADOS DA ESCOLA		
TABELA: ALUNO		
CAMPO	DESCRIÇÃO	TIPO
Nome	Nome do aluno	Caracter
Matrícula	Número que identifica o aluno na escola	Inteiro
Data_Nascimeto	Data de nascimento do aluno	Data
Sexo	Sexo do aluno	Caracter
Endereço	Endereço do aluno	Caracter
Telefone	Telefone do aluno	Inteiro
TABELA: TURMA		
CAMPO	DESCRIÇÃO	TIPO
Cod_Turma	Código que identifica as turmas	Caracter
Ano	Ano da turma	Inteiro
TABELA: PROFESSOR		
CAMPO	DESCRIÇÃO	TIPO
Nome	Nome do professor	Caracter
Cod_Professor	Número que identifica o professor	Inteiro

Telefone	Telefone do professor	Inteiro
TABELA: DISCIPLINA		
CAMPO	DESCRIÇÃO	TIPO
Nome	Nome da disciplina	Caracter
Cod_Disciplina	Número que identifica a disciplina	Inteiro

Para trabalhar com o Modelo de Entidade e Relacionamento é preciso entender os conceitos que envolvem esta técnica de modelagem. No ER, os principais conceitos são entidade, atributo, relacionamento e generalização/especialização.

Entidade

Este conceito é o mais importante na abordagem do ER. Elmasri et al. (2005) o define como sendo “um objeto com uma existência física (por exemplo, uma pessoa, um carro, uma casa ou um funcionário) ou um objeto com uma existência conceitual (por exemplo, uma empresa, um trabalho ou um curso universitário)”. Isto é, este conceito é o nome da entidade pelo qual foi definida no modelo de entidade e relacionamento.

A entidade representa objetos concretos da realidade humana, podendo ser, conforme cita o autor anteriormente um imóvel, uma pessoa e um automóvel; como também pode representar objetos abstratos, assim como um endereço. No diagrama, ela é representada na forma de um retângulo com o nome da entidade, conforme a Figura 2.5.



Figura 2.5: Representação gráfica de entidades

Cada retângulo representará um conjunto de objetos que se deseja guardar as informações. Sendo assim, na Figura 2.5, o primeiro retângulo trata-se do conjunto com

todas as informações a respeito dos alunos matriculados em uma determinada escola, os quais se deseja manter no seu banco de dados, enquanto o segundo retângulo trata-se de informações de cada turma que se deseja ser armazenado no banco. Sendo assim, tem-se no exemplo da Figura 2.5 uma representação gráfica de entidades diferentes.

Atributo

O atributo é todo o dado que é associado a cada ocorrência de uma entidade ou de um relacionamento. Conforme Heuser (1999) explica, “O conceito de atributo serve para associar informações a ocorrências de entidades ou relacionamentos”. Em um mesmo exemplo citado pelo autor de um diagrama, a Figura 2.6 tem uma entidade chamada PROJETO, que a cada ocorrência sua associam-se três principais atributos, indispensáveis em uma entidade, os quais são um nome, um código e um tipo.



Figura 2.6: Principais atributos de uma entidade

Em um diagrama de entidade e relacionamento, o atributo é representado pela figura da elipse (Figura 2.6), com a representação textual indicando o atributo dentro dela. A representação gráfica do modelo ER mais comunmente usada é a de forma de diagramas (ver Figura 2.7), contudo é possível realizar uma representação gráfica em formato de tabela (ver Tabela 2.8).



Figura 2.7: Atributos de uma entidade em forma de diagrama

Tabela 2.8: Atributos de uma entidade em forma de tabela

ALUNO	
	Nome: Caracter
PK	<u>Matrícula: Inteiro</u>
	Data_Nascimento: Data
	Sexo: Caracter
	Endereço: Caracter
	Telefone: Inteiro

No exemplo anterior (Figura 2.7), se pode observar a entidade ALUNO, com seis atributos, respectivamente nome, matrícula, data_nascimento, sexo, endereço e telefone.

Em um Modelo de Entidade de Relacionamento, há vários tipos de atributo, os quais podem ser classificados em simples ou compostos, univalorado ou multivalorado e armazenado ou derivado. E ainda pode ser classificado em valor *null*, isto é, atributo nulo.

Os atributos compostos, segundo Elmasri et al. (2005), são aqueles que “podem ser divididos em subpartes menores, que representam a maioria dos atributos básicos com significados independentes”. Na Figura 2.8 tem-se um atributo composto, o atributo endereço da entidade ALUNO, que pode ser subdividido em Rua, Cidade, Estado, CEP e País. Elmasri et al. (2005) ainda que enfatizam “os atributos que não são divisíveis são chamados simples ou atributos atômicos. E os atributos compostos podem formar uma hierarquia”. Isto é, até mesmo um atributo que já foi subdividido, poderá se

dividir novamente. Por exemplo, o atributo endereço da entidade ALUNO (Tabela 2.8) foi subdividido em mais cinco atributos. Um deles, a rua também poderia ser subdividida novamente em número e apartamento conforme mostra a Tabela 2.9.

Tabela 2.9: Hierarquia de atributos compostos

ALUNO	
	Nome: Caracter
PK	<u>Matrícula: Inteiro</u>
	Data_Nascimento: Data
	Sexo: Caracter
	Endereço: Caracter Rua: Cidade: Estado: CEP: País:
	Telefone: Inteiro

Os atributos chamados de monovalorados são aqueles que possuem um valor único para a entidade associada. Por exemplo, na Figura 2.8, tem-se a entidade ALUNO com o atributo telefone. Este pode ser considerado um atributo monovalorado ou multivalorado, pois irá depender de quantos números de telefone o aluno possui. Caso o aluno possua um único número de telefone, este será considerado um atributo monovalorado, isto é, possui um único valor, entretanto se o aluno possuir dois ou três números de telefone, o atributo possuirá dois ou três valores, sendo assim, um atributo multivalorado.

Quanto a atributo armazenados e atributos derivados, Sanches (2005) o explica:

Alguns atributos podem ter uma relação entre si. Por exemplo, *idade* e *data-nascimento* de uma *pessoa*. Para uma pessoa em particular, podemos determinar o valor atual de *idade* através do atributo *data-nascimento*. Então *idade* é chamado um atributo **derivado** e é derivado do atributo *data-nascimento*. Alguns atributos podem ser derivados de entidades relacionadas. Por exemplo, um atributo *número-empregados* de uma entidade *departamento* pode ser derivado através da contagem de número de empregados que *trabalham-para* um *departamento*. (Sanches, 2005)

Logo se percebe ao observar a Figura 2.10 que o atributo idade da entidade PESSOA é derivado do atributo data_nascimento, que por sua vez é um atributo armazenado. O atributo derivado é representado através de uma elipse com contorno tracejado no diagrama.

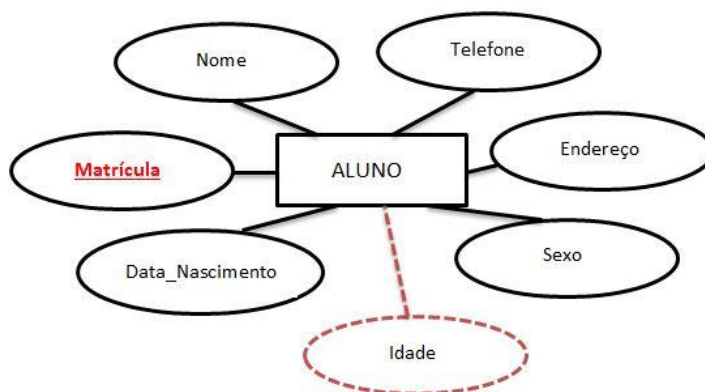


Figura 2.10: Atributo derivado

Há também casos, em que uma entidade possui atributos que não possuem nenhum valor aplicável e quando acontece isso, o atributo é chamado de *null*, isto é, nulo. Por exemplo, na Figura 2.7, a entidade ALUNO possui seis atributos associados: o nome, matrícula, data_nascimento, sexo, endereço e telefone. Este último atributo, em alguns casos, poderá ser considerado um atributo *null*, já que algum aluno pode não ter número de telefone, então este campo torna-se desconhecido, isto é, o valor do atributo existe, mas está faltando.

Cardinalidade

Uma outra característica importante do atributo é quanto à sua cardinalidade, conforme Heuser (1999) trata:

Um atributo pode possuir uma cardinalidade, de maneira análoga a uma entidade em um relacionamento. A cardinalidade de um atributo define quantos valores deste atributo podem estar associados a uma ocorrência da entidade/relacionamento a qual pertence. A representação diagramática da cardinalidade de atributos é derivada da representação da cardinalidade de entidades de relacionamentos. (Heuser, 1999, p. 23-24).

Cardinalidades de Atributos

Há casos em que o diagrama não apresenta a cardinalidade sobre os atributos. Isto só acontece quando a cardinalidade é (1,1), sendo assim ela pode ser omitida do

diagrama. Na Figura 2.11, tem-se um exemplo de cardinalidade, onde os atributos nome, matrícula, data_nascimento, sexo e endereço são considerados campos obrigatórios, por possuir uma cardinalidade mínima “1”, isto é, a entidade possui no mínimo um valor associado a entidade. Entretanto, o último atributo, telefone, pode ser considerado um atributo opcional, sendo assim possui uma cardinalidade mínima “0” e uma cardinalidade multivalorada, ou seja, uma cardinalidade máxima “n”.

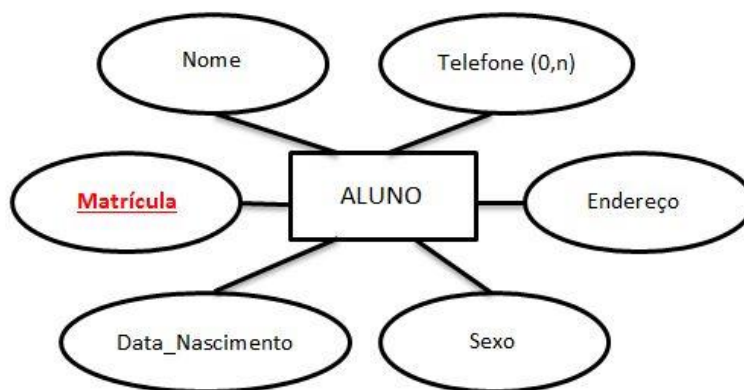


Figura 2.11: Cardinalidade de atributos no diagrama

Atributo-Chave ou Identificador

Cada entidade possui um atributo-chave ou identificador. Heuser (1999) o explica como “um identificador é um conjunto de um ou mais atributos cujos valores servem para distinguir uma ocorrência de entidade das demais ocorrências da mesma entidade”. A Figura 2.11 mostra um exemplo de um atributo-chave em um diagrama, o qual é identificado como sublinhado e cada entidade possui um único atributo-chave. Nela, a entidade ALUNO está associada a seis atributos, sendo que o identificado como atributo-chave é a matrícula, pois na escola poderá haver inúmeros alunos matriculados, mas o número de matrícula de cada aluno será diferente, a qual se torna o identificador do aluno na escola.

Relacionamento

Sobre relacionamentos de um ER, Heuser (1999) afirma que “além de especificar os objetos sobre os quais se deseja manter informações, o modelo ER deve permitir a especificação das propriedades dos objetos que serão armazenadas no BD.”

Sendo assim, relacionamento pode ser tratado como sendo um conjunto de todas as associações entre as entidades do banco de dados.

No diagrama, o relacionamento é representado na forma de um losango, sendo ligado através de linhas nos retângulos que representam as entidades no modelo. Na Figura 2.12 observa-se um diagrama de entidade e relacionamento que contém duas entidades e um relacionamento.



Figura 2.12: Representação gráfica de relacionamentos

Cardinalidade de relacionamentos

A cardinalidade de relacionamentos é uma propriedade importante no projeto do banco de dados, pois ela estabelece a quantidade de ocorrências de uma determinada entidade que pode estar associada a uma determinada ocorrência através do relacionamento. A esta propriedade, dá-se o nome de cardinalidade de uma entidade de relacionamento (Heuser, 1999).

A cardinalidade de relacionamentos entre entidades esta dividida em duas partes: a cardinalidade máxima e a cardinalidade mínima.

Cardinalidade máxima

Para explicar o conceito de cardinalidade, tem-se a Figura 2.13 que expressa uma representação gráfica dos relacionamentos com a cardinalidade.



Figura 2.13: Cardinalidade de relacionamentos

A Figura 2.13, mostra a entidade ALUNO com uma cardinalidade máxima 1 no relacionamento INSCRIÇÃO. Isto significa que uma ocorrência da entidade ALUNO pode estar associada a, no máximo uma ocorrência da entidade CURSO, ou melhor, que um aluno pode estar inscrito em no máximo um curso. Já a entidade CURSO tem cardinalidade máxima n (muitos) no relacionamento de INSCRIÇÃO. Isso significa que uma ocorrência da entidade CURSO pode estar associada a, no máximo, n ocorrências da entidade ALUNO; ou ainda, que em um curso podem se inscrever inúmeros alunos.

Para fins práticos, não é necessário distinguir entre diferentes cardinalidades máximas maiores que 1. Por esse motivo, apenas duas cardinalidades máximas são relevantes: a cardinalidade máxima **1** e a cardinalidade máxima “muitos”, referida pela letra **n**. (Heuser, 1999, p.16)

Assim, o exemplo da Figura 2.13 mostra que a cardinalidade máxima da entidade CURSO no relacionamento INSCRIÇÃO é **n**. Todavia que a cardinalidade é sempre expressa a uma ocorrência da entidade ao lado oposto da notação.

Contudo, a cardinalidade máxima ainda pode ser classificada quanto aos seus relacionamentos, podendo ser considerados relacionamentos binários ou ternários.

O relacionamento binário é aquele onde as suas ocorrências possuem duas ocorrências de entidades. Ainda há uma classificação dentre o relacionamento binário, os quais podem ser 1:1 (um-para-um), 1:n (um-para-n) e n:n (muitos-para-muitos). Nas figuras 2.14, 2.13 e 2.15 apresentam respectivamente tais exemplos.



Figura 2.14: Relacionamento 1:1 (um-para-um)

Neste tipo de relacionamento 1:1, conforme a Figura 2.14 mostra o exemplo do relacionamento CASAMENTO, em que um marido pode possuir no máximo uma mulher e uma esposa pode possuir no máximo um marido.

No relacionamento 1:n (um-para-muitos), tem-se o exemplo já explicado anteriormente da figura 2.15.



Figura 2.15: Relacionamento n:n (muitos-para-muitos)

No relacionamento n:n, tem-se o exemplo da Figura 2.15, que onde mostra um relacionamento de consulta entre o médico e o paciente. Para isso, o médico efetua consultas com muitos pacientes, da mesma forma que o paciente pode ter consultas com muitos médicos.

E ainda, o relacionamento CASAMENTO do exemplo dado na Figura 2.14, também é classificado como um relacionamento binário. Segundo Heuser (1999), “o que determina o fato de o relacionamento ser binário é o número de ocorrências de entidade que participam de cada ocorrência do relacionamento”. Sendo assim, para cada ocorrência do relacionamento CASAMENTO, há exatamente duas ocorrências de entidade MARIDO e MULHER, por isso o relacionamento é considerado binário.

No caso de relacionamentos de grau maior que dois, o conceito de cardinalidade de relacionamento é uma extensão não trivial do conceito de cardinalidade em relacionamentos binários. Lembre-se que, em um relacionamento binário R entre duas entidades A e B, a cardinalidade máxima de A em R indica quantas ocorrências de B podem estar associadas a cada ocorrência de A. No caso de um relacionamento ternário, a cardinalidade refere-se a *pares de entidades*. Em um relacionamento R entre três entidades A, B e C, a cardinalidade máxima de A e B dentro de R indica quantas ocorrências de C podem estar associadas a um par de ocorrências de A e B. (Heuser, 1999, p.19-20)

O mesmo pode ser observado no exemplo da Figura 2.16, que expressa o que o autor explica quanto a relacionamentos ternários. Nesse caso, como não é determinada a cardinalidade, vamos denotá-la genericamente de “x”.

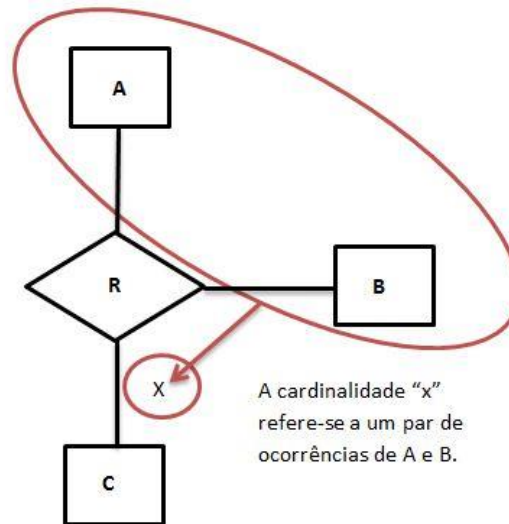


Figura 2.16: Cardinalidade de relacionamento ternário

Cardinalidade mínima

Assim como há uma cardinalidade máxima nos relacionamentos, há também uma mínima. São consideradas apenas duas cardinalidades mínimas, a cardinalidade mínima 0 e a cardinalidade mínima 1. A cardinalidade mínima 0 é denominada associação opcional e a cardinalidade mínima 1 de associação obrigatória, já que a mesma deve indicar um relacionamento obrigatório que associará a uma ocorrência de entidade.

Conforme mostra a Figura 2.17, a cardinalidade mínima é anotada ao diagrama juntamente à cardinalidade máxima. Este exemplo serve para especificar que a cardinalidade mínima, onde cada professor deve ter alocado a ele obrigatoriamente uma mesa, isto é, uma cardinalidade mínima 1, e que a mesa independente de estar ou não alocado por um professor pode existir.

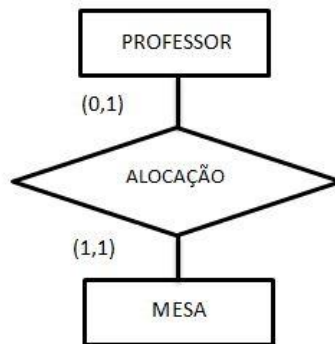


Figura 2.17: Cardinalidade mínima de relacionamento

Generalização/Especialização

A generalização/especialização são também, assim como os relacionamentos e atributos, propriedades que podem ser descritas no modelo ER.

A especialização é o processo de classificar uma classe de objetos em subclasses mais especializadas. A generalização é o processo inverso de generalizar diversas classes em uma classe abstrata de nível mais alto que inclua os objetos de todas essas classes. A especialização é um refinamento conceitual, enquanto a generalização é uma síntese conceitual. As subclasses são usadas no modelo ER para representar a especialização e a generalização. (Elmasri et al., 2005, p.79)

Através deste conceito, é possível atribuir propriedades particulares a um subconjunto das ocorrências de uma entidade considerada genérica (Heuser, 1999). Para representar a generalização/especialização é utilizado um triângulo isóscele. O exemplo da Figura 2.18 mostra a entidade CLIENTE que está subdividida em outras duas entidades: PESSOA FÍSICA e PESSOA JURÍDICA, as quais tem suas propriedades particulares. A esses dois novos subconjuntos do atributo CLIENTE, se denominam subclasses. E ainda, a entidade PESSOA FÍSICA possui, além de seus atributos CIC e sexo, os atributos associados da entidade CLIENTE, isto é, conta também com os atributos nome e código.

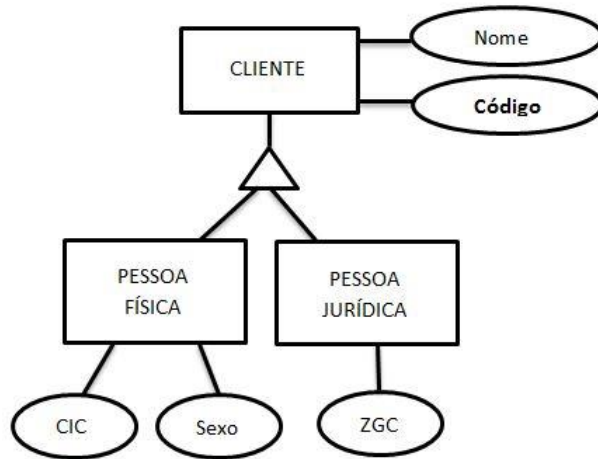


Figura 2.18: Generalização/especialização

A generalização/especialização é classificada em total ou parcial, e irá depender da entidade possuir ou não obrigatoriedade de ligação com a entidade considerada genérica. Para Heuser (1999), “Em uma generalização/especialização total para cada ocorrência da entidade genérica existe sempre uma ocorrência em uma das entidades especializadas”. Esse tipo de generalização/especialização é representado pela letra “t”, conforme mostra a Figura 2.19. Assim, ao observar este símbolo na Figura 2.19, entende-se que todo CLIENTE é PESSOA FÍSICA ou PESSOA JURÍDICA.

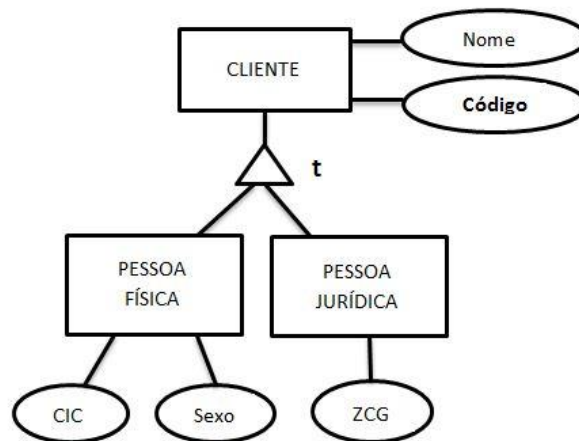


Figura 2.19: Generalização/especialização total

Na generalização/especialização parcial é quando não há uma obrigatoriedade de ocorrência da entidade genérica com a entidade especializada. Conforme a Figura 2.20, que ilustra um exemplo de generalização/especialização parcial, onde a entidade FUNCIONÁRIO possui duas ocorrências de entidades genéricas: MOTORISTA e

SECRETÁRIA, mas nem todo funcionário é motorista ou secretária. A este tipo de generalização/especialização tem-se a representação pela letra “p”.

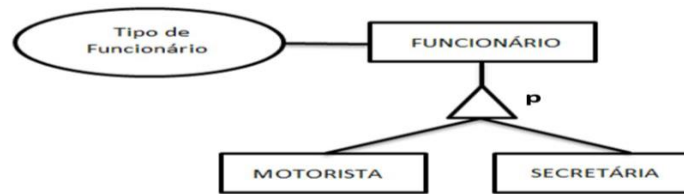


Figura 2.20: Generalização/especialização parcial

Ainda cabe lembrar que não há um número limite de níveis hierárquicos da generalização/especialização. Segundo Heuser (1999), “é admissível, inclusive, que uma mesma entidade seja especialização de diversas entidades genéricas (a chamada *herança múltipla*)”. Isto é, uma subclasse pode ser descendente de mais de uma superclasse. Para melhor explicação, usou-se o mesmo exemplo do autor sobre herança múltipla. Na Figura 2.21, há a entidade BARCO, onde a mesma é uma especialização de um VEÍCULO AQUÁTICO e que este é uma especialização de VEÍCULO. Sendo assim, o barco além de suas próprias propriedades, este também apresenta as propriedades do veículo aquático e as de um veículo no geral. Logo, a herança múltipla é apresentada na entidade VEÍCULO ANFÍBIO, pois este além de possuir as suas próprias propriedades específicas, possui propriedades das entidades VEÍCULO TERRESTRE e VEÍCULO AQUÁTICO, já que o mesmo é especialização dessas duas entidades.

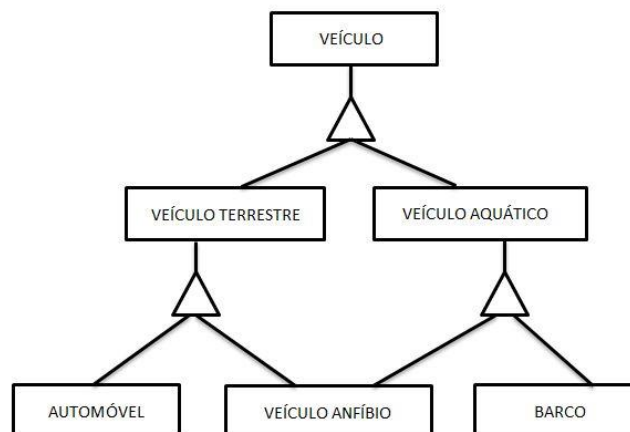


Figura 2.21: Generalização/especialização com herança múltipla

Diagrama Entidade Relacionamento

Conforme já foi apresentado anteriormente, o Diagrama Entidade Relacionamento (diagrama ER) é a principal ferramenta do modelo conceitual Entidade-Relacionamento, já que é a sua representação gráfica. Segundo Rodrigues (2014), “Em situações práticas, o diagrama é tido muitas vezes como sinônimo de modelo, uma vez que sem uma forma de visualizar as informações, o modelo pode ficar abstrato demais para auxiliar no desenvolvimento do sistema”. Sendo assim, quando da modelagem dos dados, torna-se importante já criar a representação gráfica do projeto.

Para a criação do diagrama ER (ver Figura 2.22), optou-se pela utilização de tabelas, já que um banco de dados relacionais é constituído tanto por tabelas e quanto por relações. Como o conceito de tabela é um conjunto de linhas não ordenadas, onde cada linha é composta por inúmeros campos, e estes identificados por um nome de campo, os mesmos que formam uma coluna (Figura 2.22). Assim este diagrama é formado por três tabelas: clientes, encomendas e artigos, as quais estão se relacionando. O que possibilita este relacionamento, segundo Heuser (1999) é “o conceito básico para identificar linhas e estabelecer relações entre linhas de tabelas de dados relacionais é a *chave*”. Dentre esses dois tipos de chave, consideram-se duas muito importantes: a chave primária e a chave estrangeira.

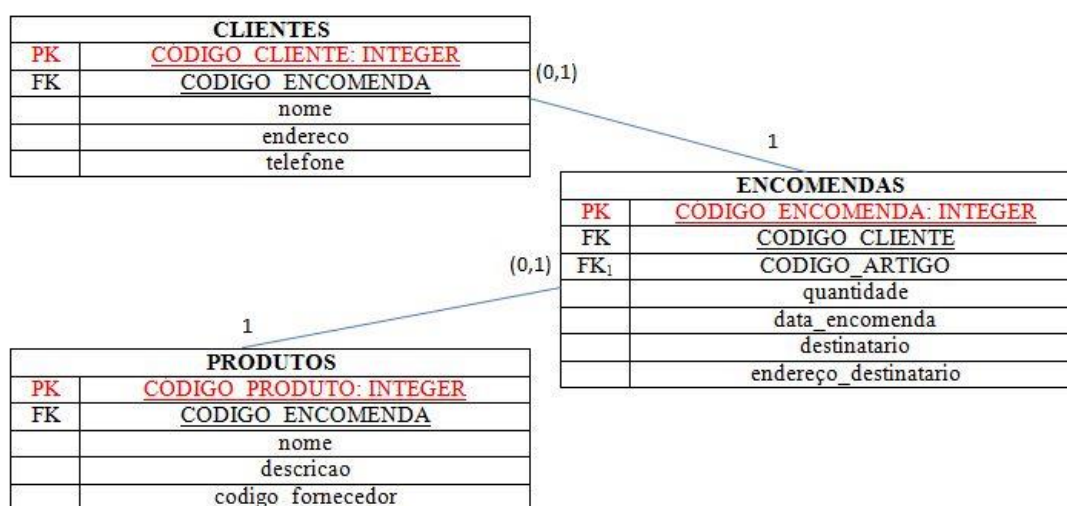


Figura 2.22: Relações entre tabelas

A chave primária é uma identificação única que obrigatoriamente a tabela deve conter, podendo ser classificada em simples ou composta. A chave primária simples é quando a tabela possui um único campo que identifique exclusivamente cada registro armazenado, isto é, não pode haver nela dois ou mais registros com o mesmo valor para esse campo. Já a chave composta, segundo Gonçalves (2011) é “formada por mais de um campo, os valores de cada campo podem se repetir, mas não a combinação desses valores”. A esta chave tem-se a notação “PK”, por ser a sigla da palavra inglesa “Primary Key”.

A chave estrangeira, segundo Heuser (1999) “é uma coluna ou uma combinação de colunas, cujos valores aparecem necessariamente na chave primária de uma tabela. A chave estrangeira é o mecanismo que permite a implementação de relacionamentos em um banco de dados relacional”. Isto é, sempre que houver uma relação entre as tabelas, as mesmas terão que ter a chave estrangeira. Ela permite que os campos não sejam repetidos, caso uma tabela tenha a necessidade de utilizar algum campo de uma outra tabela, a chave estrangeira é utilizada para relacionar as tabelas e não ter que repetir os campos. Esta chave é representada por “FK”, sigla da palavra “Foreign Key”.

Conforme a Figura 2.22, a tabela produtos tem colunas código_cliente (chave primária), nome, endereço e telefone. As linhas que ligam as tabelas indicam que há um relacionamento entre elas. A chave estrangeira (FK) indica as colunas que fazem parte do relacionamento. Sendo assim, todas as tabelas possuem chaves primárias e estrangeiras, o que impede que os campos sejam repetidos.

3. METODOLOGIA

Para atingir os objetivos propostos, a Figura 3.1 apresenta o fluxograma da metodologia adotada para a realização deste trabalho, dividida em quatro etapas subsequentes.

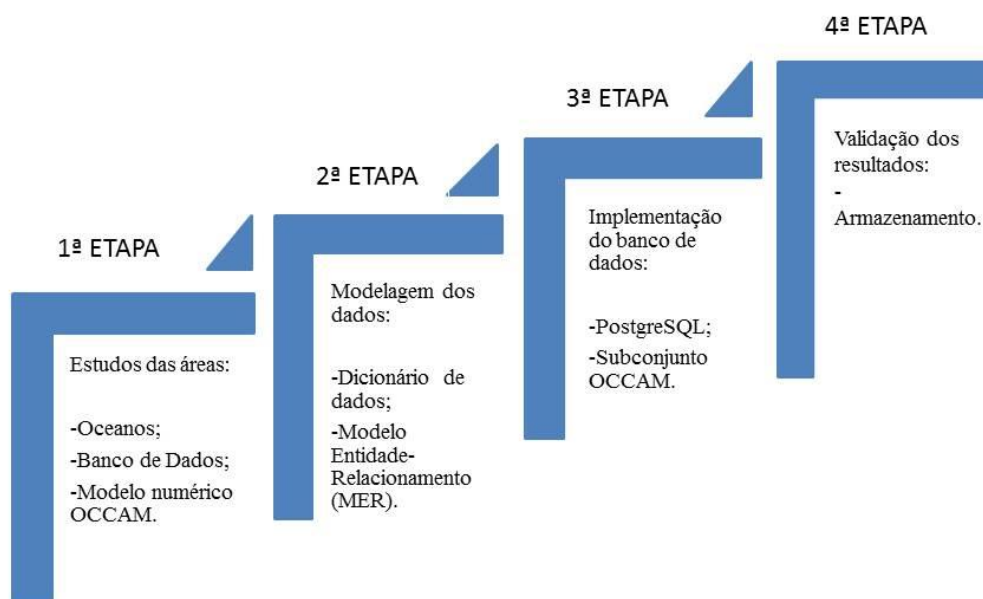


Figura 3.1: Metodologia dividida em quatro etapas

A primeira etapa consistiu em estudos dirigidos aos conceitos oceanográficos e de sistemas de banco de dados. Primeiramente, ocorreu o contato com uma rotina de coleta dos dados oceanográficos do trabalho, conseqüentemente houve a necessidade do estudo deste contexto, o qual ocorreu por meio de livros e pesquisas na internet do mundo que envolve a Oceanografia e o seu principal objeto de estudo: o oceano. Após isso, foi realizado estudo sobre os sistemas de dados, já que um dos objetivos principais do trabalho é desenvolver um banco de dados. Sendo assim, o foco dessa parte de estudo foram todos os conceitos de entidade de relacionamento, banco de dados e o Sistema Gerenciador de Banco de Dados PostgreSQL. E por fim, se estudou o modelo

numérico primitivo OCCAM (*Ocean and Circulation Advanced Modeling*), onde os dados foram extraídos do próprio site do OCCAM. Estes dados que foram armazenados junto ao grupo de pesquisa de Oceanografia Física da FURG, foram utilizados na realização de diversos tipos de pesquisas.

A segunda etapa foi a realização de um mapeamento de todas as variáveis presentes no modelo OCCAM, para que assim fosse possível elaborar o modelo ER, procurando encontrar os relacionamentos entre as variáveis. Mas antes de elaborar o modelo ER, foi organizado um dicionário de dados, com a finalidade de explicar todos os objetos presentes do modelo em forma textual. A criação do dicionário de dados auxiliou na criação de um modelo relacional na forma de um modelo conceitual.

A terceira e quarta etapa do trabalho foram as mais trabalhosas e que exigiram inúmeras pesquisas. A terceira etapa consistiu na conversão das informações para a base de dados. Para isso, houve a escolha de um SGBD (Sistema Gerenciador de Banco de Dados), o *PostgreSQL*. Esta escolha deve-se ao fato de este ser considerado o SGBD de código aberto mais utilizado pela comunidade científica, que conta com consultas completas, chaves estrangeiras, facilidade de acesso, entre outros, possibilitando desenvolver um banco de dados relacionais. Sendo assim, foi preciso criar as tabelas conforme o modelo ER desenvolvido na terceira etapa com suas relações e chaves estrangeiras e estrangeiras. Logo após, foi preciso extrair os arquivos da rotina de dados do modelo OCCAM que se encontra em formato NetCDF, ou como usualmente é chamado de “nc” no formato “txt”. Para extraí-los foi necessário abrir os dados no MATLAB, mas antes disso, houve a necessidade de primeiro instalar um pacote “nctoolbox” adicional ao programa para visualizar os dados de um dia de rotina do OCCAM. Depois do “nctoolbox” instalado no MATLAB, foi inserido neste programa um único dia de rotina de dados, sendo o do primeiro dia de coleta, isto é, a rotina do dia 1 de janeiro do ano de 1988. Ao visualizar os dados no MATLAB, a segunda parte dessa terceira etapa foi a inclusão desses dados no SGBD *PostgreSQL*.

Após inserção dos dados no *PostgreSQL*, foi executado a quarta etapa, que consistiu na análise dos resultados a respeito de seu armazenamento. Já que uma das finalidades deste trabalho também é diminuir a repetição e replicação dos dados, os quais aumentam o espaço de armazenamento. E o armazenamento no SGBD

PostgreSQL diminui este armazenamento, já que trabalha com o menor número de variáveis, e trabalha com o modelo de banco de dados relacional, não precisando repetir as variáveis anteriormente declaradas.

4. MODELO PROPOSTO

As rotinas de coleta dos dados do modelo OCCAM que são utilizados no Laboratório de Oceanografia Costeira e Estuarina da FURG diz a respeito a um conjunto de dados que inicia no dia 1 de janeiro de 1988 ao dia 31 de dezembro de 2004, com resolução temporal de cinco dias, totalizando ao longo desses 16 anos 1168 coletas de 24 variáveis (o tamanho médio de um dia de coleta é de 432 Megabytes). Mas para a realização do projeto de banco de dados oceanográficos foi validado apenas um dia de coleta de dados do modelo OCCAM, uma vez que o primeiro dia de coleta esteja validado, o mesmo poderá ser repetido para os demais, já que se trata de rotinas de mesma variáveis com datas diferentes de coletas. E para que isso possa ser criado, foi necessário desenvolver a modelagem conceitual e o projeto lógico desse trabalho, os quais serão apresentados a seguir.

4.1 Dicionário de Dados

Para o desenvolvimento da modelagem conceitual deste trabalho, foi elaborado o dicionário de dados (Tabela 4.1) que foi armazenado no banco de dados para que assim possa realizar a construção do diagrama entidade-relacionamento. Esta modelagem se tornou importante na criação do banco de dados, pois permite a organização dos termos de armazenamento de dados para a sua implementação.

O dicionário de dados é apresentado com suas identificações na tabela 4.1, onde na coluna CAMPO estão listadas as variáveis que serão inseridas no banco de dados. Já na coluna DESCRIÇÃO é exposta descrição de cada variável, com a unidade de medida que é utilizada. As outras duas colunas são as colunas TAMANHO e TIPO, onde respectivamente é identificado a dimensão de cada variável e o tipo pelo qual ela está armazenada.

Tabela 4.1: Dicionário de dados do Modelo OCCAM

DICIONÁRIO DE DADOS OCCAM			
CAMPO	DESCRIÇÃO	TAMANHO	TIPO
Day	Indica o dia da coleta de dados. A coleta é realizada	1	Inteiro

	em um período de 5 em 5 dias.		
Depth	Indica a distância entre superfície e o fundo oceânico. Unidade utilizada é em centímetros (cm).	66x1	Real
KMT	É a batimetria oceânica que constitui a entrada para o modelo com valores inteiros que representam os níveis de profundidade.	721x601	Real
Latitude T	Indica a distância ao Equador medida ao longo do meridiano de Greenwich. Esta distância mede-se em graus, podendo variar entre 0° e 90° para Norte(N) ou para Sul(S). A Latitude T significa que as suas coordenadas são em graus.	601x1	Real
Latitude U	Indica a distância ao Equador medida ao longo do meridiano de Greenwich. A Latitude U significa que as suas coordenadas são UTM, ou seja, a posição Global em metros.	601x1	Real
Longitude T	Indica a distância ao meridiano de Greenwich medida ao longo do Equador. Esta distância mede-se em graus, podendo variar entre 0° e 180° para Leste(E) ou para Oeste(W). A Longitude T significa que as suas coordenadas são em graus.	721x1	Real
Longitude U	Indica a distância ao meridiano de Greenwich medida ao longo do Equador. A Longitude U significa que as suas coordenadas são UTM, ou seja, a posição global em metros.	721x1	Real
Max Mixed Layer Depth Mean	Profundidade média da camada de mistura máxima. Unidade utilizada é em centímetros (cm).	721x601	Real
Min Mixed Layer Depth Mean	Profundidade mínima da camada de mistura máxima. Unidade utilizada é em centímetros (cm).	721x601	Real
Mixed Layer Mean Depth	Profundidade média da camada de mistura m. Unidade utilizada é em centímetros (cm).	721x601	Real
Potential Temperature Mean	É a temperatura potencial média de uma parcela de ar. A unidade utilizada é em graus Celsius (°C).	721x601x66	Real

Runoff Mean	Escoamento médio. Unidade utilizada é a unidade específica de navegação milha náutica por segundo (nm/s).	721x601	Real
Salinity Mean	É definida como peso em gramas dos sais dissolvidos em 1 quilograma de água do mar. A salinidade varia de 34 a 37, com média próxima de 35 PSU. A unidade utilizada é PSU (Escala Prática de Salinidade).	721x601x66	Real
Sea Surface Height Mean	Indica a altura média da superfície do mar. Unidade utilizada é em centímetros (cm).	721x601	Real
Seconds	Indica os segundos da coleta.	1	Inteiro
Total Freshwater Into Ocean M	Aporte total de água doce no oceano. Unidade utilizada é a unidade específica de navegação milha náutica por segundo (nm/s).	21x601	Real
Timestep	Passo de integração, por indicar o número de coletas realizadas. Uma localização temporal.	1	Inteiro
U Velocity Mean	Componente U da velocidade média da corrente. A unidade utilizada é centímetros por segundo (cm/s).	721x601x66	Real
U Wind Stress Mean	Componente U do médio do vento. A unidade utilizada é em Pascal (Pa).	721x601	Real
V Velocity Mean	Componente V da velocidade média da corrente. A unidade utilizada é centímetros por segundo (cm/s).	721x601x66	Real
V Wind Stress Mean	Componente V do stress médio do vento. A unidade utilizada é em Pascal (Pa).	721x601	Real
Years	Indica o ano da coleta realizada.	1	Inteiro

Ao observar este dicionário, pode-se perceber o quanto um dia apenas de dados torna o banco de dados volumoso, devido a alta quantidade de dados. Um exemplo disso é o terceiro campo do dicionário, a variável KMT, com 721x601 de tamanho, que irá gerar um total de 433.321 dados. Isso acontece, porque esta variável coleta dados conforme a combinação da Latitude T com a Longitude T, logo para cada Longitude T

(com 721 longitudes diferentes) haverá 601 latitudes diferentes. E isso se repete pelas demais variáveis listadas conforme as suas dimensões.

E vale também ressaltar que o total de variáveis apresentadas em um único dia de rotina é de 24 variáveis. Mas neste dicionário apresentado não foi introduzido duas variáveis: a DEPTH_EDGES e KMU. Estas não foram apresentadas, pois ao longo dos estudos de cada variável não foi possível entender a sua descrição e sendo assim estas não foram colocadas no banco de dados.

4.2 Modelo Entidade-Relacionamento e Diagrama Entidade-Relacionamento

Conforme já foi visto no capítulo 2, o Modelo Entidade-Relacionamento (MER) trata-se do modelo conceitual que descreverá todas as entidades, atributos com os seus relacionamentos. Ele torna-se necessário, pois a partir deste modelo que foi possível entender como as entidades se relacionam entre si. Para que isso aconteça, foram analisados as dimensões dos dados oceanográficos, pois só através delas foi possível entender os relacionamentos entre as entidades.

O MER é apresentado na Figura 4.2, onde a primeira entidade foi denominada ITERAÇÃO, onde foram agrupados os dados da coleta, armazenando o dia, ano e segundos. A partir dela, foram criadas as outras duas entidades que se relacionam com a entidade ITERAÇÃO, onde ambas são as entidades chamada COORDENADAS T e COORDENADAS U. Nessas duas entidades são armazenadas as duplas de latitudes e longitudes de coordenadas T e U. Como as latitudes possuem dimensões 601x1 e as longitudes dimensões 721x1, logo foram agrupadas aos demais variáveis em entidades que também possuíam dimensões que envolvessem tanto as latitudes quanto as longitudes. Logo, isso aconteceu nas próximas entidades criadas. Partindo disso, as próximas entidades relacionadas às entidades das coordenadas T ou U, foram respectivamente, DADOS T e DADOS U com variáveis de dimensões 721x601. As outras duas entidades criadas são para armazenar as variáveis que possuíam três dimensões, isto é, além das dimensões das latitudes/longitudes, havia também a dimensão da profundidade de 66 níveis. Logo, as variáveis com dimensões 721x601x66 foram divididas nas entidades DADOS COMPLEMENTARES T e DADOS

COMPLEMENTARES U, onde elas eram armazenadas conforme as suas dimensões e as quais coordenadas pertenciam T ou U. E por fim, a última entidade criada foi a da PROFUNDIDADE que se relaciona com as entidades DADOS COMPLEMENTARES T e DADOS COMPLEMENTARES U, já ela possui dimensão 66x1.

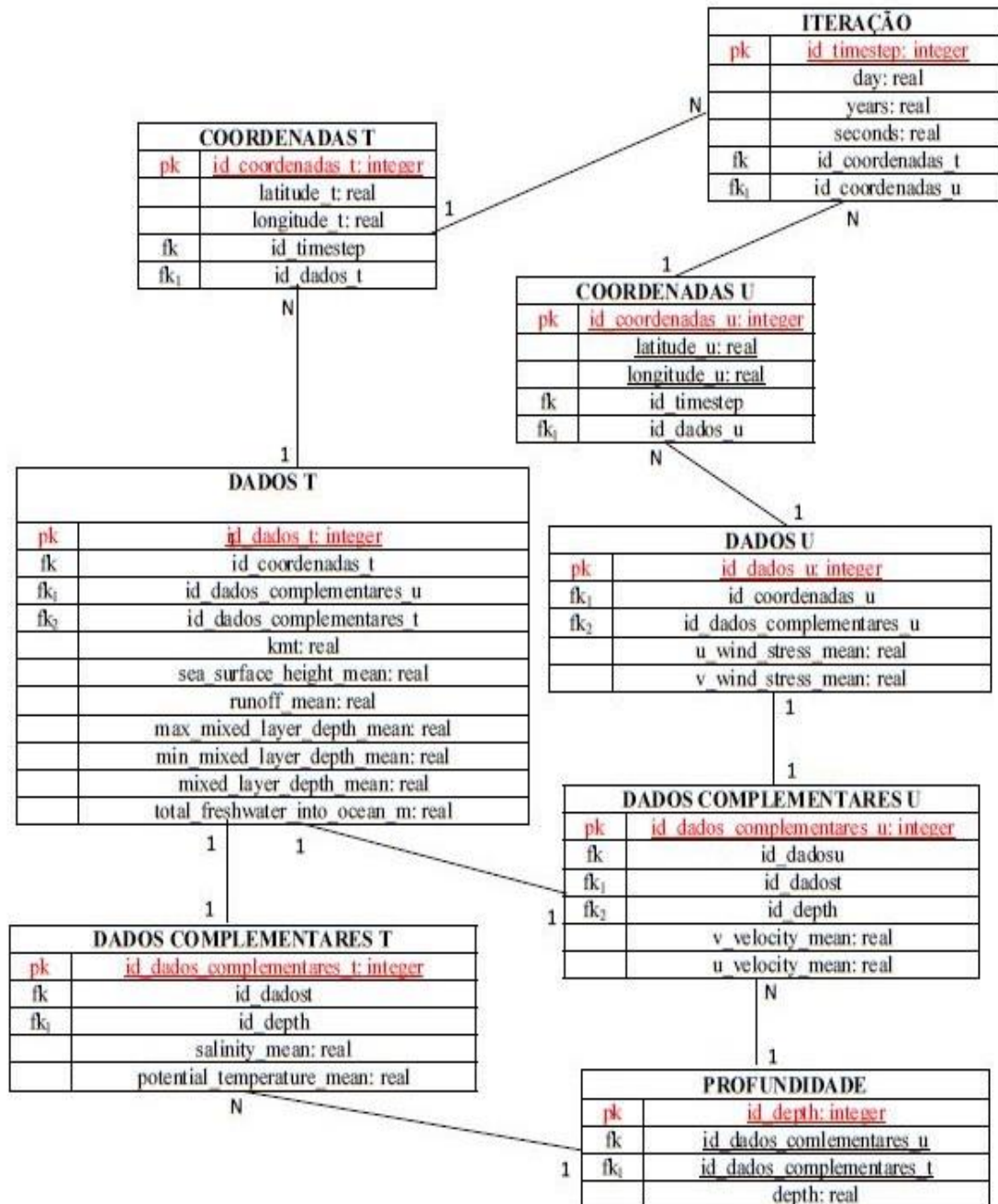


Figura 4.2: Modelo ER dos dados oceanográficos em representação de tabelas

O diagrama ER acima (Figura 4.2), é representado por oito tabelas com seus atributos, relacionamentos, chaves primárias e estrangeiras e suas cardinalidades. Sendo assim, as tabelas são: Iteração, Coordenadas T, Coordenadas U, Dados T, Dados U, Dados Complementares T, Dados Complementares U e Profundidade. E em cada tabela, são definidos seus atributos listados, que são as variáveis dos dados do modelo OCCAM com o seu respectivo tipo e as chaves primária e estrangeira quando a possuem. Para melhor explicar, será analisado a cada tabela.

Na primeira tabela contém a tabela Iteração com atributos: day, years e seconds, todos do tipo real. Esta tabela possui três chaves, a chave primária (PK) representada por id_timestep do tipo inteiro e duas chaves estrangeiras (FK): id_coordenadas_t e id_coordenadas_u. Ela possui essas duas chaves estrangeiras, pois ela se relaciona com a tabela COORDENADAS T e COORDENADAS U, onde as chaves primárias de ambas tornam-se chaves estrangeira da tabela ITERAÇÃO. E o mesmo acontece com as demais tabelas que se relacionam, onde todas possuem uma única chave primária e recebem chaves estrangeiras das tabelas pelas quais se relacionam. Isso acontece para que uma tabela possa se relacionar com a outra sem que precise repetir nenhuma variável que já foi instanciada na outra tabela com a qual está se relacionando.

A segunda tabela denominada COORDENADAS_T possui dois atributos: longitude_t e latitude_t, ambas do tipo real. Possuem a chave primária id_coordenadas_t do tipo inteiro e como possui dois relacionamentos entre tabelas, possui duas chaves estrangeiras: id_timestep e id_dados_t.

A terceira tabela é a COORDENADAS_U com atributos: latitude_u e longitude_u, ambas do tipo real. Sua chave primária é o id_coordenadas_u do tipo inteiro e duas chaves estrangeiras: id_timestep e id_dados_u.

A quarta tabela denominada DADOS T com todos os seus atributos do tipo real, os quais são: kmt, sea_surface_height_mean, runoff_mean, max_mixed_layer_depth_mean, min_mixed_layer_depth_mean, mixed_layer_depth_mean e total_freshwater_into_ocean_m. Como possui três relacionamentos com as tabelas COORDENADAS T, DADOS COMPLEMENTARES U e DADOS

COMPLEMENTARES T, logo possui três chaves estrangeiras: id_coordenadas_t, id_dados_complementares_u e id_dados_complementares_t.

A quinta tabela é chamada DADOS U com os seus atributos do tipo real que são: u_wind_stress_mean e v_wind_stress_mean. A sua chave primária do tipo inteiro é a id_dados_u e as chaves estrangeiras são id_coordenadas_u e id_dados_complementares_u.

A sexta tabela é a DADOS COMPLEMENTARES T com dois atributos do tipo real que são: salinity_mean e potential_de_temperature_mean. Sua chave primária do tipo inteiro é id_dados_complementares_t. Esta tabela possui dois relacionamentos com as tabelas DADOS T e PROFUNDIDADE, por isso possui duas chaves estrangeiras: id_dados_t e id_depth.

A sétima tabela, chamada DADOS COMPLEMENTARES U, possui dois atributos, v_velocity_mean e u_velocity_mean, ambos do tipo real. Esta tabela possui três relacionamentos DADOS T, DADOS U e PROFUNDIDADE, sendo três as suas chaves estrangeiras: id_dados_t, id_dados_u e id_depth. E sua chave primária é do tipo inteiro que é o id_dados_complementares_u.

A última tabela é a PROFUNDIDADE com um único atributo depth, do tipo real e com chave primária do tipo inteiro que é o id_depth. Como a tabela PROFUNDIDADE possui dois relacionamentos entre DADOS COMPLEMENTARES U e DADOS COMPLEMENTARES T, logo suas chaves estrangeiras são id_dados_complementares_u e id_dados_complementares_t.

Baseado nessa modelagem, percebe-se que quando uma tabela se relaciona com outra, a sua chave estrangeira será a que é chave primária da entidade do seu relacionamento. E ainda existem os relacionamentos entre as tabelas a partir de suas cardinalidades. Os relacionamentos da tabela ITERAÇÃO com as COORDENADAS U e COORDENADAS T é de 1:n. Nele tem-se que a tabela ITERAÇÃO tem cardinalidade máxima n (muitos) com relação as tabelas COORDENADAS U e COORDENADAS T, onde ambas tem cardinalidade máxima 1. Isso significa que a tabela ITERAÇÃO esta associada a n ocorrências tanto das tabelas COORDENADAS U e COORDENADAS T. E ainda, as últimas tabelas estão associadas a uma única ocorrência com a ITERAÇÃO. Logo, isto acontece já que a cardinalidade é sempre

expressa a uma ocorrência da tabela ao lado oposto da notação. E existem também outras tabelas que possuem cardinalidades 1:1, isto significa que ela esta associada a uma, e no máximo uma, ocorrência da tabela que está se relacionando.

4.3 Projeto Lógico

Após o modelo conceitual ser desenvolvido, a próxima etapa foi o desenvolvimento do projeto lógico que consiste na implementação dos dados no *PostgreSQL*. A escolha dele deve-se ao fato de que é um SGBD objeto relacional de código aberto mais avançado nesse meio, que permite vários recursos com uma facilidade de acesso. Foi desenvolvido pelo *PostgreSQL Global Development Group* com o seu lançamento em 1 de maio de 1995. Atualmente está disponível em 24 idiomas e possuem versões para grandes sistemas operacionais como o Windows, Linux e Mac OS X.

No decorrer do desenvolvimento dessa parte do trabalho, foi preciso fazer o download desse SGBD, que está disponível no site: www.postgresql.org/download/ . Ao realizar o download e instalação do *PostgreSQL*, no seu primeiro acesso necessitou de uma senha de entrada (Figura 4.3), cadastrada pelo usuário. Logo após, foi necessário a criação de um novo *DataBase* (Figura 4.4) com o nome do banco onde serão inseridos os dados oceanográficos. Este banco foi denominado OCCAM (Figura 4.5).

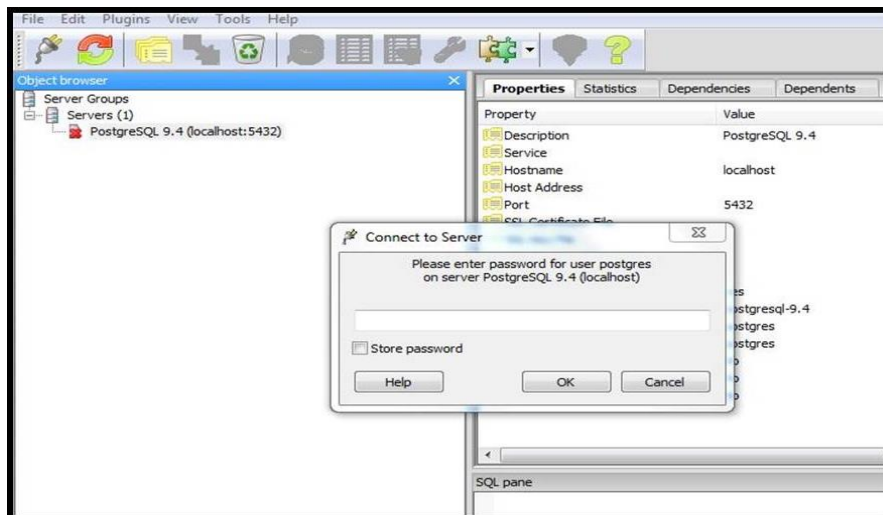


Figura 4.3: Acesso com senha

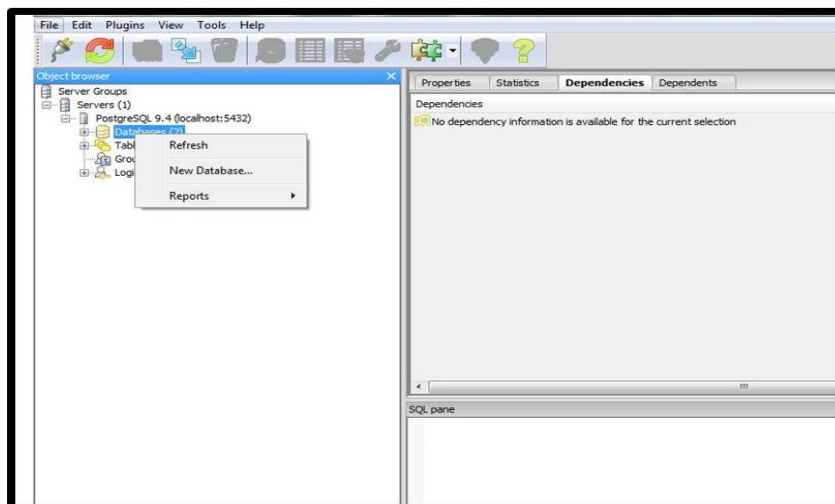


Figura 4.4: Criação de um novo *DataBase*

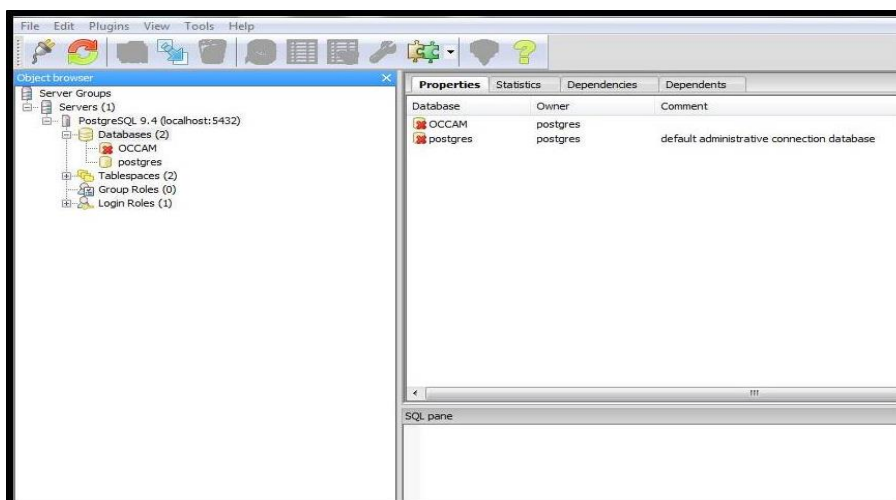


Figura 4.5: Criação do banco OCCAM

Após a criação do banco OCCAM no *PostgreSQL*, a próxima etapa foi a criação das oito tabelas, cada qual, com atributos, chaves primárias e estrangeiras e seus relacionamentos, tudo baseado no modelo ER. Ao criar as tabelas (Figura 4.6), foi gerado um script para cada uma, com todas as informações de cada uma delas.

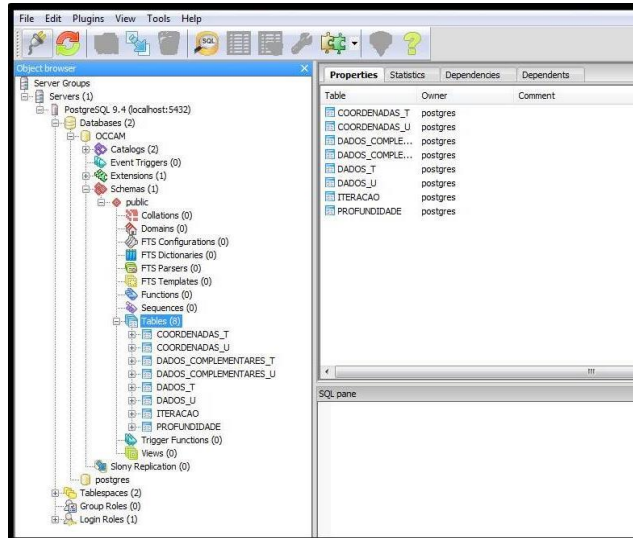


Figura 4.6: Criação das oito tabelas

Para cada tabela criada no banco, foi gerado um script individual com informações dos atributos, tipo e chaves. Sendo assim, foram criados oito scripts. Para entendimento, a Figura 4.7 apresenta o script da tabela ITERAÇÃO. Os outros scripts estão disponíveis no Anexo 1.

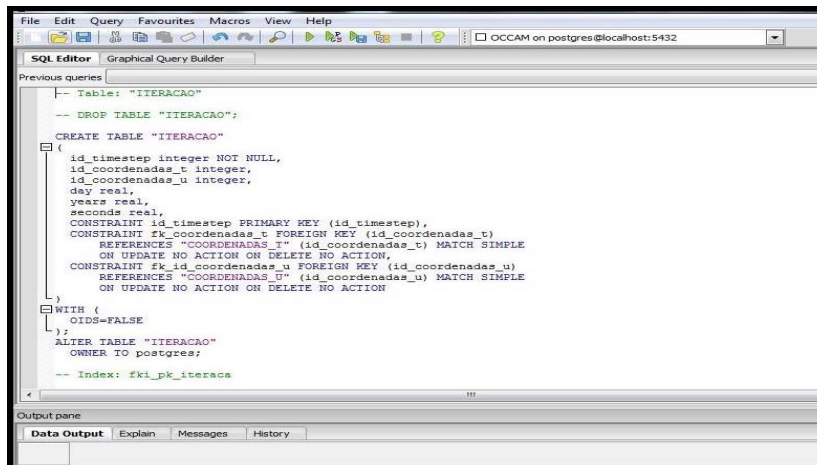


Figura 4.7: Script da tabela ITERAÇÃO

Percebe-se no script de criação da tabela ITERAÇÃO da Figura 4.7 que ela possui seis colunas com os campos respectivos id_timestep, id_coordenadas_t e id_coordenadas_u, day, seconds e years. Todos os seis campos com o tipo, conforme o modelo ER. Nele também está a declaração das chaves primária e estrangeira.

Identifica-se que o `id_timestep` é a chave primária (Primary key), a chave estrangeira é a `fk_id_coordenadas_t`, a mesma que é pertencente a tabela COORDENADAS T. Outra chave estrangeira presente nesta tabela é a `fk_id_coordenadas_u`, a qual é pertencente a tabela COORDENADAS U.

4.4 Obtenção dos dados

Ao desenvolver a modelagem conceitual, através do dicionário de dados, a criação do modelo ER do banco de dados e a implementação do último no SGBD *PostgreSQL*, o próximo passo foi a obtenção dos dados oceanográficos. Como os dados estão salvos em arquivos compactados no formato *NetCDF*, o mais lógico a fazer seria exportar esses dados através do MATLAB, já que este software era o utilizado pelos pesquisadores do Laboratório de Oceanografia para visualizar os dados, pois este permite interações de alto desempenho destinadas ao cálculo numérico, e ainda, no seu sistema interativo tem como elemento básico de informação uma matriz sem exigir o seu dimensionamento.

Para dar seqüência ao trabalho, foi necessário que o MATLAB fosse instalado e ainda instalar o pacote “*nctoolbox*” adicionalmente ao programa para visualizar os dados de um dia de rotina do OCCAM.

Sendo assim, a próxima etapa a ser concluída seria na visualização dos dados de um dia de rotina no programa. Para que isso acontecesse, foi preciso definir quais comandos deveriam ser feitos para a sua visualização e extração, criando uma rotina no MATLAB. Essa etapa tomou muito tempo e dedicação, já que não encontravam formas para gerar uma rotina de comandos para que os dados fossem gerados no software. Ela tornou-se muito difícil, pois na maioria das vezes, as pesquisas com demonstrações de comandos encontradas se tornavam inúteis e sem sucesso na prática. Após muita insistência e pesquisas, foi criada uma rotina de comandos para que os dados oceanográficos fossem abertos e analisados.

Para abrir os dados oceanográficos foi digitado no Editor de Linhas de Comando do MATLAB o comando “`ncdisp`”, com o local do arquivo de um dia de coleta de dados, logo sendo este o comando dado: `ncdisp('01_01_88__0001.nc')` . Foi utilizado este comando, pois no *ncdisp* irá conter as informações essenciais sobre as variáveis e as suas dimensões do arquivo denominado `01_01_88_0001.nc` . Sendo

assim, foram geradas no MATLAB todas as informações das 24 variáveis, suas dimensões, tipos e atributos, conforme mostra a Figura 4.8.

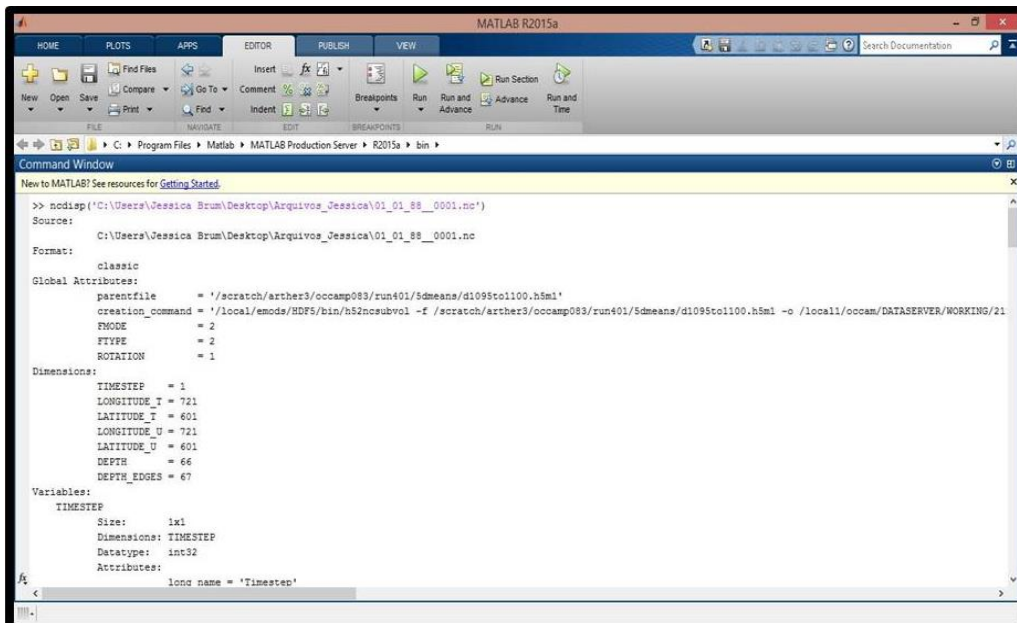


Figura 4.8: Informações geradas ao declarar o comando *ncdisp* no MATLAB

Analisando os dados apresentados através deste comando na Figura 4.8, percebe-se que no campo “Dimensions” ela lista sete variáveis, as quais são: TIMESTEP, LONGITUDE_T, LATITUDE_T, LONGITUDE_U, LATITUDE_U, DEPTH e DEPTH_EDGES, com dimensões, respectivamente, de 1, 721, 601, 721, 601, 66 e 67. Mas vale lembrar que as variáveis KMU e DEPTH_EDGES não serão analisadas nesse trabalho, conforme foi explicado no item 4.1 deste capítulo.

O próximo campo que se pode analisar é o “Variables” ainda na Figura 4.8, em que ela lista todas as informações de cada variável. Como se pode ser visto, a primeira a ser listada é a variável TIMESTEP, com tamanho 1x1 e do tipo de dado no valor inteiro imutável.

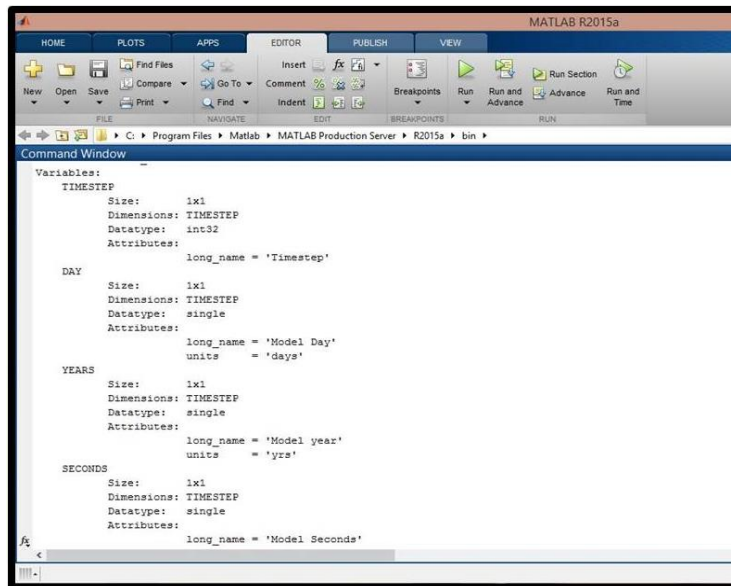


Figura 4.9: Informações geradas ao declarar o comando *ncdisp* no MATLAB

Na Figura 4.9 estão a continuação das variáveis que estão presentes no arquivo 01_01_88_0001.nc após o comando *ncdisp*. Como a variável *TIMESTEP* já foi explicada anteriormente, a análise será das demais apresentadas nessa figura. E estas são as variáveis *DAY*, *YEARS* e *SECONDS*. Elas são de tamanho 1x1, com dimensões *TIMESTEP*, ou seja, dimensão 1 e são tipos de dados únicos.

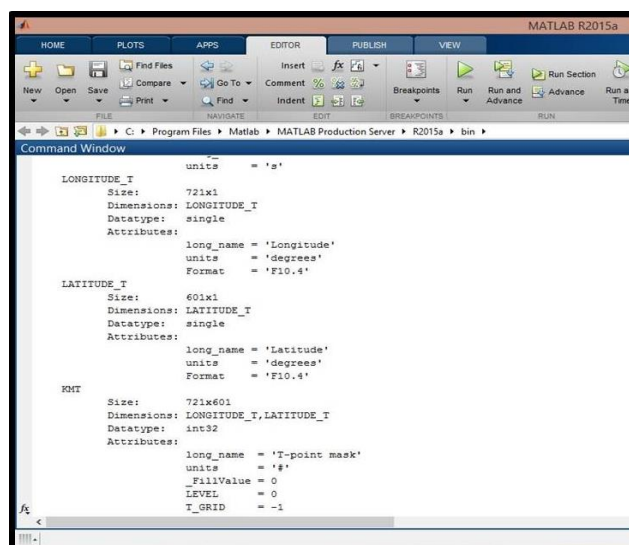


Figura 4.10: Informações geradas ao declarar o comando *ncdisp* no MATLAB

As próximas variáveis apresentadas na Figura 4.10 são *LONGITUDE_T* e *LATITUDE_T*, são de tamanhos respectivamente de 721x1 e 601x1. Estas são do tipo de dados únicos (single), enquanto *KMT* um tipo de dado no valor inteiro imutável e com dimensões *LATITUDE_T*, *LONGITUDE_T*, isto é 721x601.

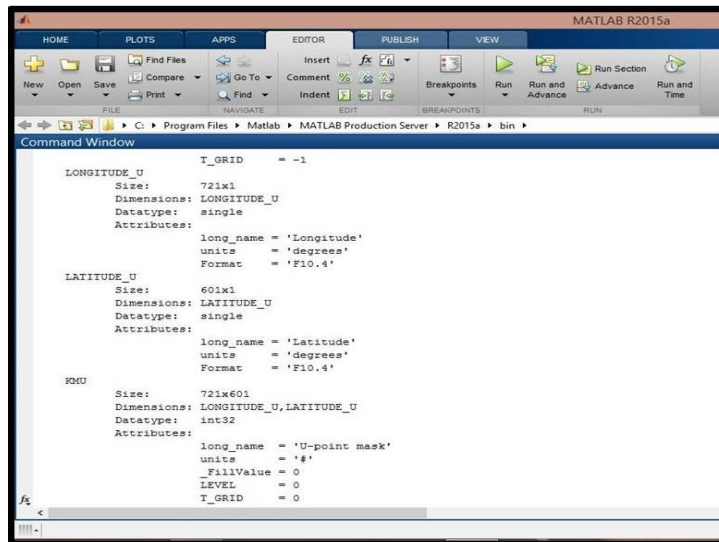


Figura 4.11: Informações geradas ao declarar o comando *ncdisp* no MATLAB

A Figura 4.11 mostra as variáveis `LONGITUDE_U` e `LATITUDE_U` com tamanhos respectivamente de `721x1` e `601x1`, com tipo de dados únicos.

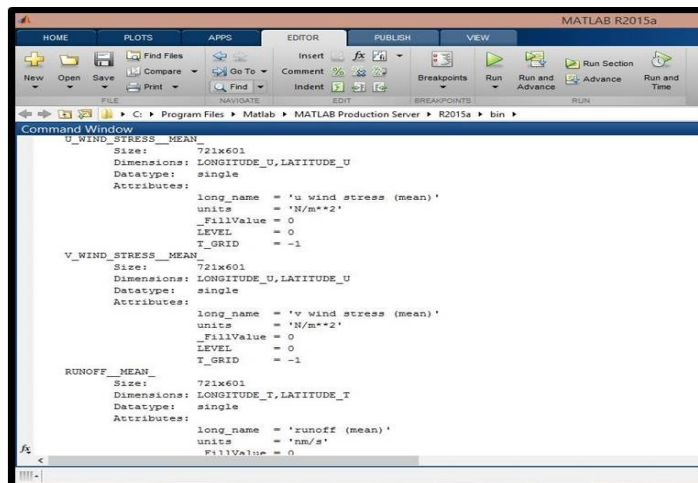


Figura 4.12: Informações geradas ao declarar o comando *ncdisp* no MATLAB

Na Figura 4.12 estão mais três variáveis: `U_WIND_STRESS_MEAN_`, `V_WIND_STRESS_MEAN_` e `RUNOFF_MEAN_`, cada uma com um tipo de dados no valor inteiro imutável. As duas primeiras variáveis possuem dimensões `LONGITUDE_U`, `LATITUDE_U`, isto é, com dimensões `721x601`. E a última variável possui dimensão `LONGITUDE_T`, `LATITUDE_T`, ou seja, dimensão `721x601`.

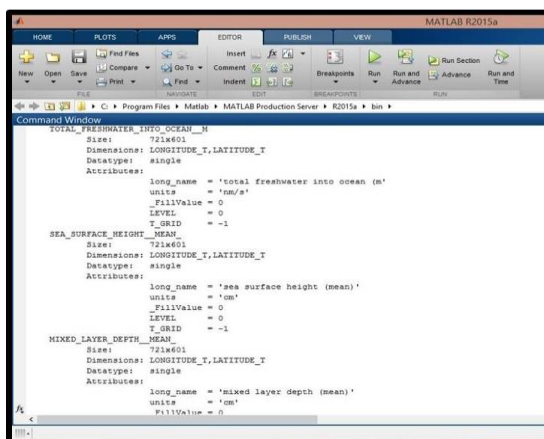


Figura 4.13: Informações geradas ao declarar o comando *ncdisp* no MATLAB

As variáveis `TOTAL_FRESHWATER_INTRO_OCEAN_M`, `SEA_SURFACE_HEIGHT_MEAN` e `MIXED_LAYER_DEPTH_MEAN` (Figura 4.13) possuem as mesmas dimensões e tipo de dado, tais como respectivamente 721x601 e valor inteiro imutável.

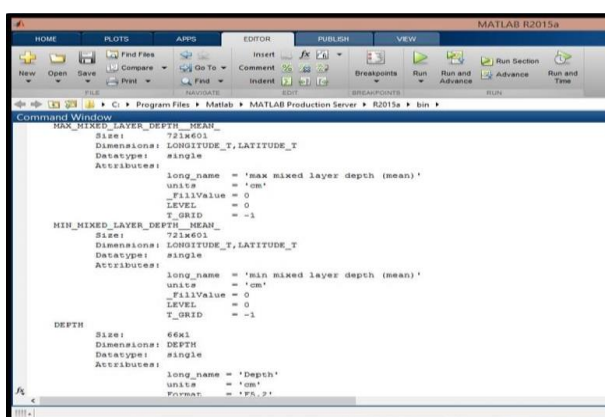


Figura 4.14: Informações geradas ao declarar o comando *ncdisp* no MATLAB

Outras três variáveis apresentadas na figura 4.14 são `MAX_MIXED_LAYER_DEPTH_MEAN` e `MIN_MIXED_LAYER_DEPTH_MEAN`, ambas de dimensões 721x601 (LONGITUDE_T, LATITUDE_T) e de tipo de valor inteiro imutável. E ainda a variável `DEPTH` com dimensão DEPTH, ou seja, dimensão 66x1 e também tipo de valor inteiro imutável.

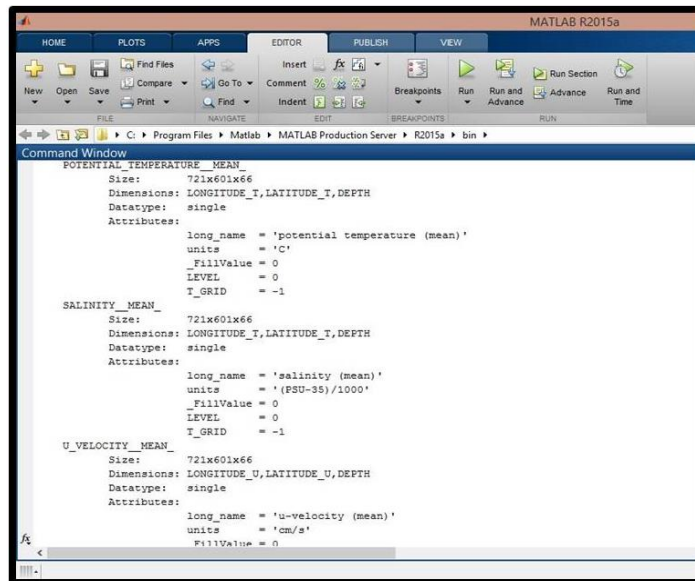


Figura 4.15: Informações geradas ao declarar o comando *ncdisp* no MATLAB

As três variáveis (Figura 4.15) *POTENTIAL_TEMPERATURE_MEAN_*, *SALINITY_MEAN_* e *U_VELOCITY_MEAN_*, estas com três dimensões e tipo de valor inteiro imutável. As duas primeiras variáveis com dimensões (LONGITUDE_T, LATITUDE_T, DEPTH) 721x601x66 e a última com dimensão (LONGITUDE_U, LATITUDE_U, DEPTH).

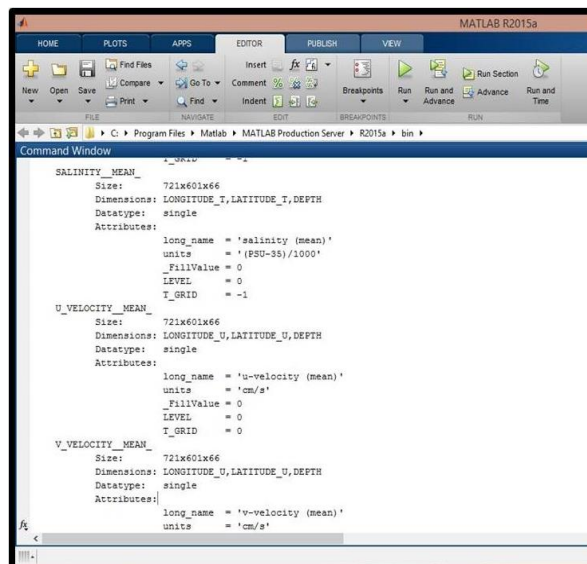


Figura 4.16: Informações geradas ao declarar o comando *ncdisp* no MATLAB

Por fim, a última variável é a `V_VELOCITY_MEAN_` de dimensão (`LONGITUDE_U`, `LATITUDE_U`, `DEPTH`) `721x601x66` e também do tipo de valor inteiro imutável.

Outro comando que foi usado para abrir os dados é mostrado na Figura 4.17. O “`ncid1`” abre um arquivo *NetCDF* chamado '`01_01_88__0001.nc`' e atribui um número de arquivo para isto. Este número de arquivo é a única saída da função, e é colocado em nossa variável `ncid1` para usos posteriores.

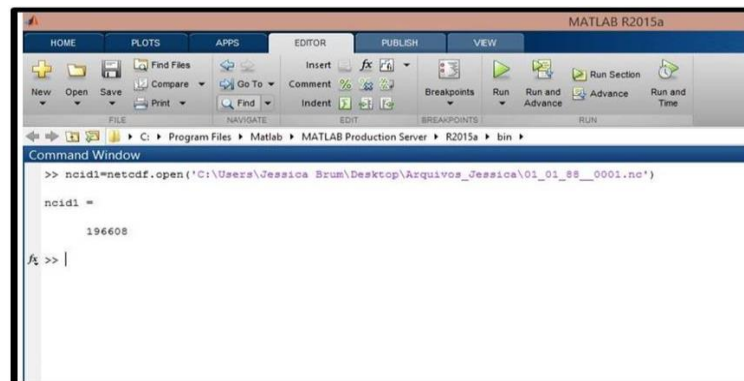


Figura 4.17: Comando `ncid1` no MATLAB

Após essa etapa, foram pesquisados e usados todos os comandos necessários para visualizar e entender os dados oceanográficos. E os últimos comandos da rotina são para visualizar os dados armazenados em cada variável. Toda a rotina dos comandos que foram utilizados para a visualização dos dados está disponível no anexo 2. Para isso, a Figura 4.18 apresenta o comando utilizado para abrir os dados da variável `LONGITUDE_T` e os dados gerados no MATLAB. O comando utilizado para isso foi `LONGITUDE_T=ncread ('01_01_88__0001.nc', 'LONGITUDE_T')`. Esse gera todos os dados armazenados nessa variável e armazena em uma tabela no *Workspace* do MATLAB (Figura 4.19).

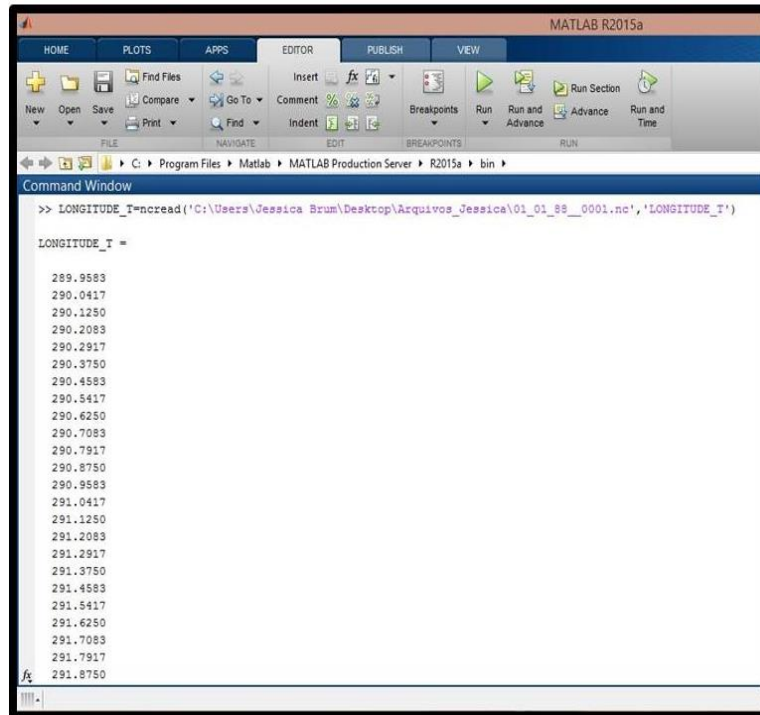


Figura 4.18: Comando para abrir os dados da variável LONGITUDE_T

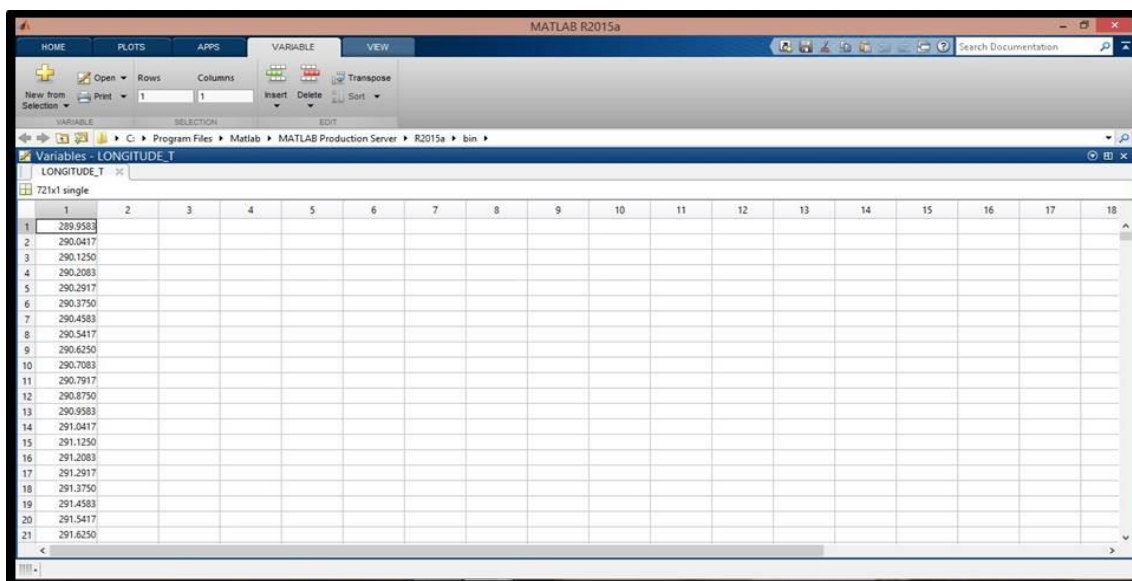


Figura 4.19: Janela do *Workspace* do MATLAB com dados armazenados na variável LONGITUDE_T

A Figura 4.19 mostra a matriz formada na janela do *Workspace* do MATLAB. Como já dito anteriormente, a dimensão da variável LONGITUDE_T é de 721x1, logo se formou a matriz 721x1 de dados dessa variável. Outra variável, KMT, foi também

visualizada pelo mesmo comando, mudando apenas o nome da variável: `KMT=ncread('01_01_88__0001.nc', 'KMT')`, o que gerou no *Workspace* do MATLAB a matriz de dimensão 721x601 com dados desta variável. Pode-se perceber que os dados dessa variável são gerados a partir da combinação de cada dado, da *LONGITUDE_T* (721) com dados da *LATITUDE_T* (601), sendo um total de 433.321 dados armazenados na variável *KMT*. Como não será possível mostrar todos esses dados gerados no *Workspace*, por causa do seu tamanho, uma parte dos dados são mostradas nas Figuras 4.20 e 4.21.

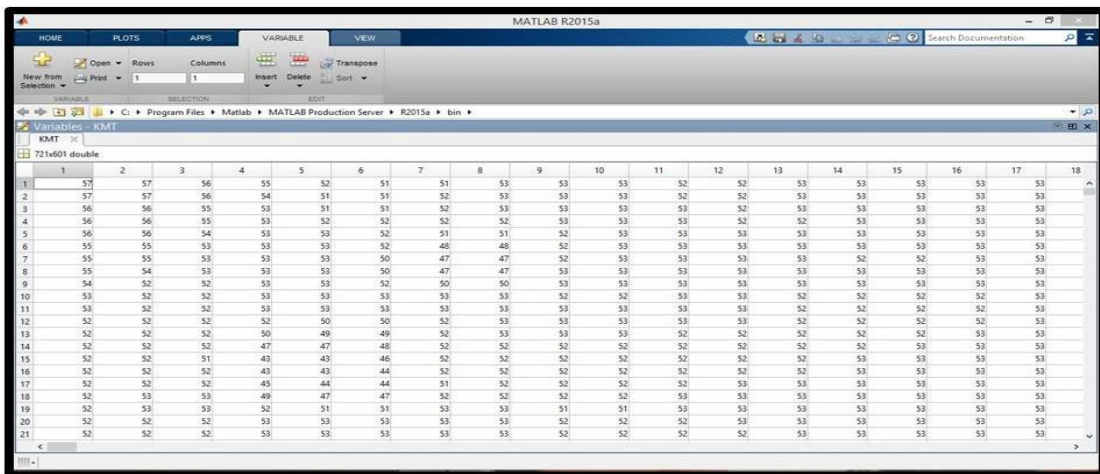


Figura 4.20: Janela do *Workspace* do MATLAB com dados armazenados na variável *KMT*

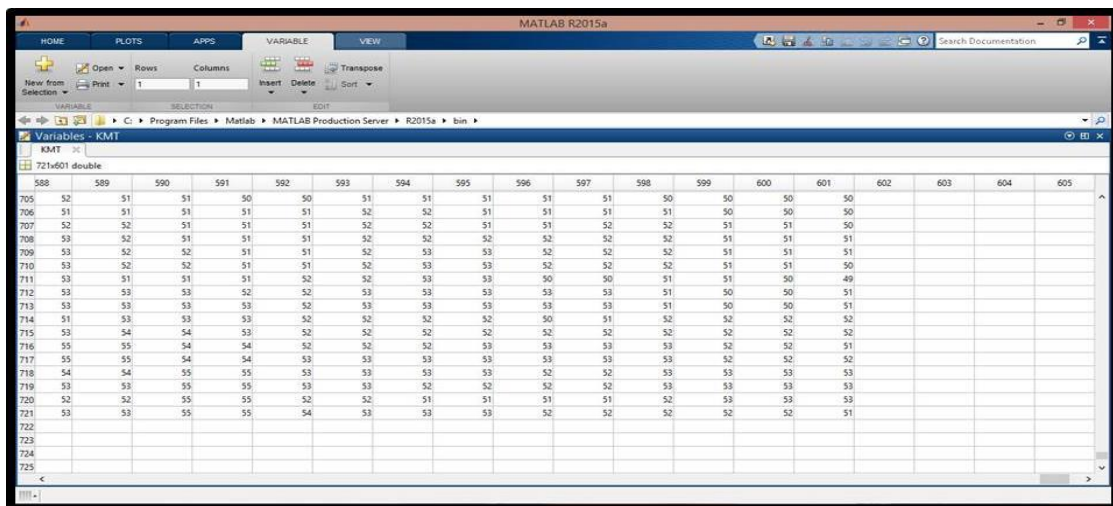


Figura 4.21: Janela do *Workspace* do MATLAB com a continuação dos dados armazenados na variável *KMT*

O mesmo comando de visualização dos dados contidos em cada variável e a geração das matrizes no *Workspace* do MATLAB serve para as demais variáveis. Apenas as variáveis com dimensões 721x601x66 gerarão matrizes tridimensionais e a estas não são possíveis ser visualizadas no *Workspace* do MATLAB, pois ele não permite a visualização dos dados das variáveis superiores a 524.148 elementos. Como a variável `U_VELOCITY_MEAN_` possui dimensão 721x601x66, o que significa que gerará 28.599.186 elementos na matriz, não é possível a sua visualização no *Workspace*, pois a quantidade de elementos é superior ao que pode ser suportado, conforme mostra a Figura 4.21.

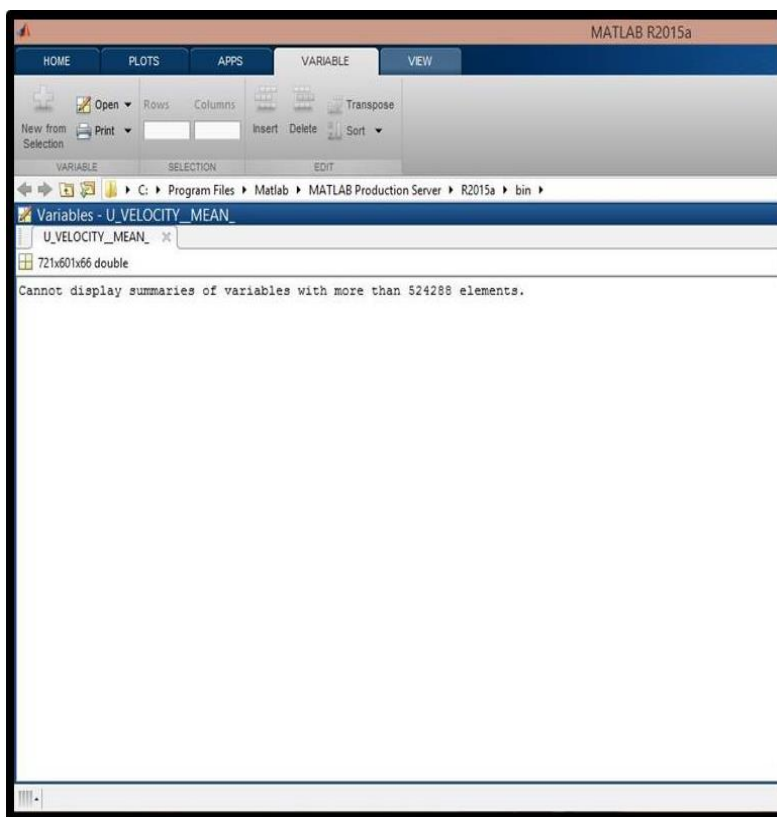


Figura 4.22: *Workspace* do MATLAB para visualizar os dados da variável `U_VELOCITY_MEAN_`

Isso também vale para as variáveis `POTENTIAL_TEMPERATURE_MEAN_`, `SALINITY_MEAN_` e `V_VELOCITY_MEAN_` com dimensões 721x601x66.

4.5 Inserção dos dados no Banco de Dados

Para popular o banco de dados foi feito um grande esforço, já que eram muitos dados a serem armazenados. O trabalho inicial seria de popular as oito tabelas presentes no *PostgreSQL*, onde cada uma delas possuem chaves primárias, as quais devem ser não nulas e únicas. Sendo assim, cada linha da tabela preenchida com um dado, necessita de uma chave que lhe identifique, sendo estas as chaves primárias. Para que isso possa ser feito e não se repita nenhuma numeração igual das chaves, ou ainda, que não se torne confuso tantas chaves diferentes, foi preciso dar um número que identifique cada tabela. Logo as chaves primárias de cada tabela foram numeradas de 1 a 8, sendo identificadas por este número, conforme mostra a Tabela 4.23.

Tabela 4.23: Tabela com o número identificador para as chaves primárias

TABELA	CHAVE PRIMÁRIA	A CHAVE COMEÇARÁ COM O NÚMERO
ITERAÇÃO	id_timestep	1
COORDENADAS T	id_coordenadas_t	2
COORDENADAS U	id_coordenadas_u	3
DADOS T	id_dados_t	4
DADOS U	id_dados_u	5
DADOS COMPLEMENTARES U	id_dados_complementares_u	6
DADOS COMPLEMENTARES T	id_dados_complementares_t	7
PROFUNDIDADE	id_depth	8

Como é visto na Tabela 4.23, todas as chaves primárias da tabela ITERAÇÃO começam com sequências de números com o dígito 1, as chaves primárias da tabela COORDENADAS_T iniciam suas sequências de números com o dígito 2, as chaves da tabela COORDENADAS_U começam com números de dígito 3 e assim sucessivamente, até a tabela PROFUNDIDADE, onde suas chaves primárias começam com sequências de números com o dígito 8.

Após ter os números que identificam a chave primária de cada tabela, o próximo passo foi a criação de cada uma das chaves ou *ids* das tabelas. Essa parte foi feita manualmente. Mas antes de tudo, foi necessário descobrir quantos *ids* deveriam conter cada tabela. Para isso, foi preciso analisar todas as tabelas e assim descobrir que deveriam conter em cada tabela 433.321 *ids*, já que nas tabelas COORDENADAS_T e COORDENADAS_U são armazenados as longitudes e latitude (T e U), logo tem-se que as longitudes afetam 721 linhas da matriz 721x1 e as latitudes afetam 601 linhas da matriz 601x1. Assim, elas gerarão, respectivamente, cada uma 721 e 601 dados. E como existem variáveis que dependem da combinação das latitudes e longitudes para gerarem a matriz com os dados, as tabelas COORDENADAS_T e COORDENADAS_U deverão conter todas as combinações de longitudes e latitudes, sendo assim uma combinação de 721x601, num total de 433.321 *ids*.

Em um arquivo texto foram digitadas sequências de *ids* para cada tabela, onde os primeiros 11 mil *ids* (chaves primárias) da primeira tabela (ITERAÇÃO) foram digitados manualmente. Depois de ter feito uma, os outros 11 mil *ids* das outras sete tabelas que restam foram consequência dos *ids* da primeira tabela, já que para isso, o que foi sendo substituída o primeiro dígito (dígito 1) de cada sequência pelo dígito que represente cada tabela. Para a criação dos demais *ids*, foi preciso digitar no arquivo de texto uma sequência de mil dígitos de três casas decimais, começando com 000 a 999 (000, 001, 002, 003, ... , 999). Depois, no mesmo arquivo foi digitado na frente de cada uma dessas sequências a letra A (A000, A001, A002, A003, ... , A999) e o arquivo foi salvo. Com o mesmo arquivo surgirá outros 25 arquivos diferentes, isto é, do primeiro arquivo foi selecionado um comando do editor de texto, e as letras A presentes neste arquivo, foram substituídas pela letra B, sempre salvando com nomes diferentes. Por fim, obteve-se 26 arquivos, já que todas as letras do alfabeto foram substituídas nesse

arquivo. Após isso, todo o conteúdo dos 26 arquivos salvos foram colocados em um único arquivo de texto, respeitando a ordem alfabética, e salvo para ser usado na criação dos *ids*. Este arquivo foi salvo com o nome MODELO. Isso foi feito, pois havia a necessidade de criação de 433.321 *ids* para as oito tabelas. Se eles fossem feitos digitando um a um, seria inviável.

Os primeiros números a serem substituídos foram para as chaves primárias da tabela ITERAÇÃO, logo no arquivo MODELO foi substituída primeiramente a letra A pelo número 100, B por 101, C por 102 e assim sucessivamente até o número 125 correspondente na substituição da última letra do arquivo, isto é, a letra Z. E sendo salvo o arquivo com nome da primeira e da última substituição, ou seja, salvar com o nome: 100mil-125mil. Esse processo foi realizado para a criação de todos os *ids* das oito tabelas, totalizando 3.466.586 *ids*.

Depois que todos os *ids* das oito tabelas foram criados, o próximo passo seria armazenar os dados oceanográficos no *PostgreSQL*. Como se tratavam de muitos dados para serem armazenados, a forma adotada para inserir os dados de um modo mais rápido foi através dos comandos das janelas do *PostgreSQL*. Para concluí-lo, foi preciso dividir este processo em três etapas. A primeira etapa consistiu em encontrar os comandos de inserção de dados a cada tabela; depois que isso foi descoberto, a segunda etapa seria de que forma inserir os inúmeros dados de cada tabela de uma forma que não comprometa muito tempo e trabalho; e por fim, conseqüentemente, a inserção dos dados no *PostgreSQL*.

Para a primeira etapa, iniciou-se com a inserção dos dados na primeira tabela, COORDENADAS_T (Figura 4.23), através das janelas de comando do programa. Antes disso, os passos anteriores foram abrir o programa, inserir a senha de acesso, abrir o banco de dados OCCAM, onde nele estão as oito tabelas. Como a primeira tabela listada é a COORDENADAS_T, logo ela foi a primeira tabela que dados correspondentes foram inseridos. Para que isso acontecesse, clicou-se com o botão direito do mouse em cima da tabela, logo foi criada a primeira janela da Figura 4.23, onde se clicou na décima linha, selecionando os Scripts e após a seleção de INSERT Scripts. Depois desses comandos, foi gerada outra janela (Figura 4.25) com um editor

de SQL para que então os dados possam ser inseridos manualmente na tabela COORDENADAS_T.

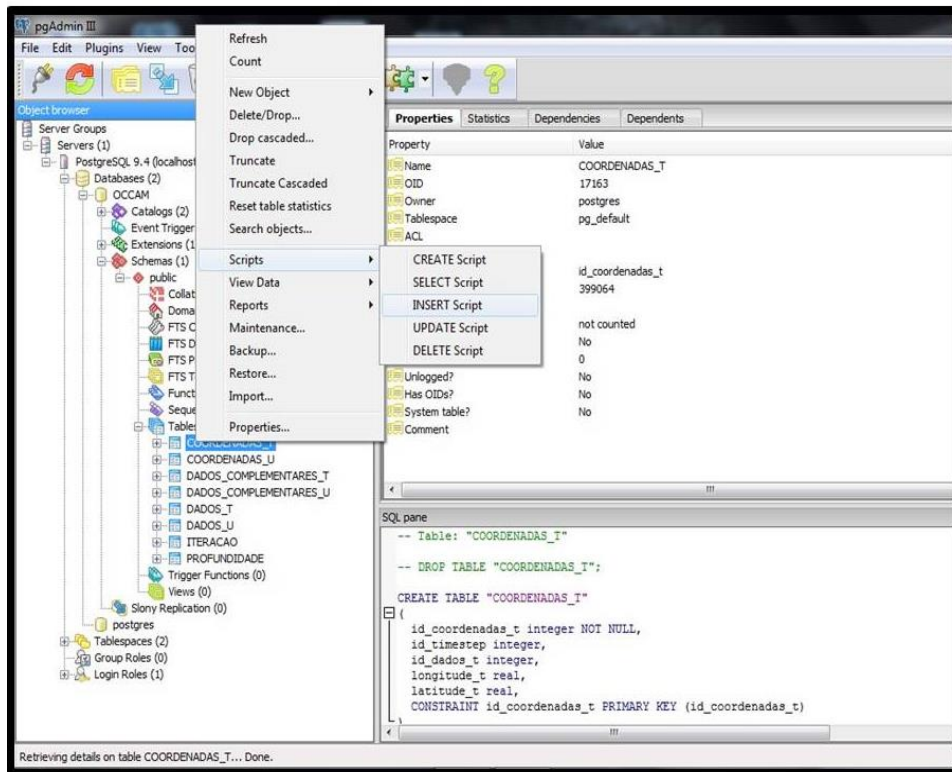


Figura 4.24: Janela de comando para inserção de dados na tabela COORDENADAS_T no PostgreSQL

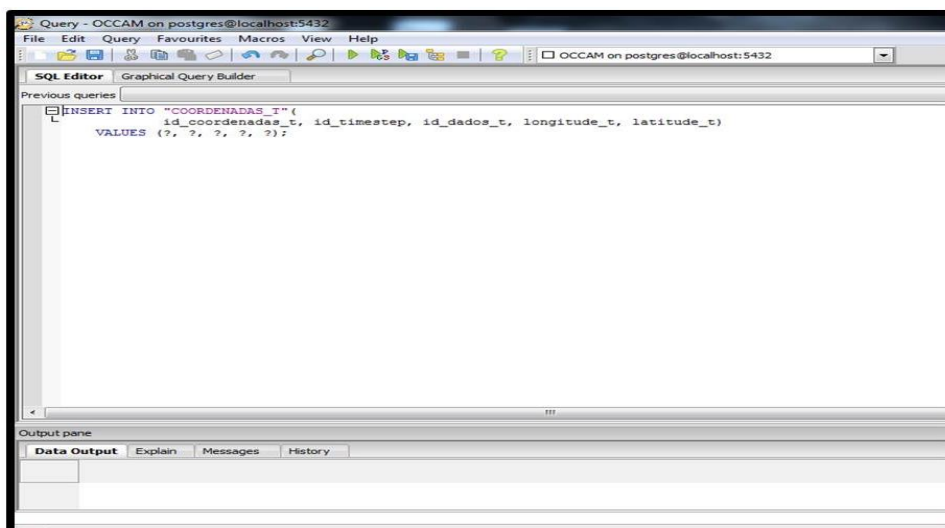


Figura 4.25: Janela do PostgreSQL para inserção dos dados na tabela COORDENADAS_T

Como pode ser observado na Figura 4.25 do SQL, para a inserção de dados da COORDENADAS_T, encontra-se na segunda linha o nome dos cinco campos presentes na tabela e na terceira linha está o espaço para serem inseridos os dados, respeitando a ordem dos campos.

Após descobrir como realizar a inserção de dados nas tabelas, a segunda etapa a ser alcançada foi de inserir muitos dados agrupados, sem que eles sejam inseridos um a um, pois isso tomaria muito tempo. Para isso foi proposto um modelo da tabela com o nome dos campos a serem inseridos na planilha (Figura 4.26) e nele preenchido em cada campo os seus respectivos dados, salvo, e após a tabela foi copiada somente com os dados e colada direto no editor do SQL da tabela (Figura 4.25) COORDENADA_T, onde devem ser inseridos os valores, e após isso apertar o botão para executar. Com isso, os dados em cada campo da tabela são armazenados. Se houver algum erro, há uma mensagem mostrando tipo e a linha de erro; caso contrário, há uma mensagem que os dados foram inseridos com sucesso na tabela. O mesmo foi feito para as outras tabelas. E após isso, podem ser visualizados os dados que foram inseridos na tabela, clicando em uma janela de visualização no painel do programa e logo será mostrada a tabela com todos os dados que já estão armazenados (Figura 4.27).

	A	B	C	D	G	H	I	J	K	L	M	N	O
1	COORDENADAS U												
2		id_coordenadas_U		id_timestep		id_dados_u		longitude_U		latitude_U			
3	(3,		1,		5,		290,		-60)			
4	(30,		10,		50,		290,		-59,9166679)			
5	(31,		11,		51,		290,		-59,8333321)			
6	(32,		12,		52,		290,		-59,75)			
7	(33,		13,		53,		290,		-59,6666679)			
8	(34,		14,		54,		290,		-59,5833321)			
9	(35,		15,		55,		290,		-59,5)			
10	(36,		16,		56,		290,		-59,4166679)			
11	(37,		17,		57,		290,		-59,3333321)			
12	(38,		18,		58,		290,		-59,25)			
13	(39,		19,		59,		290,		-59,1666679)			
14	(300,		100,		500,		290,		-59,0833321)			
15	(301,		101,		501,		290,		-59)			
16	(302,		102,		502,		290,		-58,9166679)			
17	(303,		103,		503,		290,		-58,8333321)			
18	(304,		104,		504,		290,		-58,75)			
19	(305,		105,		505,		290,		-58,6666679)			
20	(306,		106,		506,		290,		-58,5833321)			
21	(307,		107,		507,		290,		-58,5)			
22	(308,		108,		508,		290,		-58,4166679)			
23	(309,		109,		509,		290,		-58,3333321)			
24	(310,		110,		510,		290,		-58,25)			
25	(311,		111,		511,		290,		-58,1666679)			

Figura 4.26 Tabela preenchida na planilha com os dados a serem inseridos no SQL da tabela COORDENADAS_T

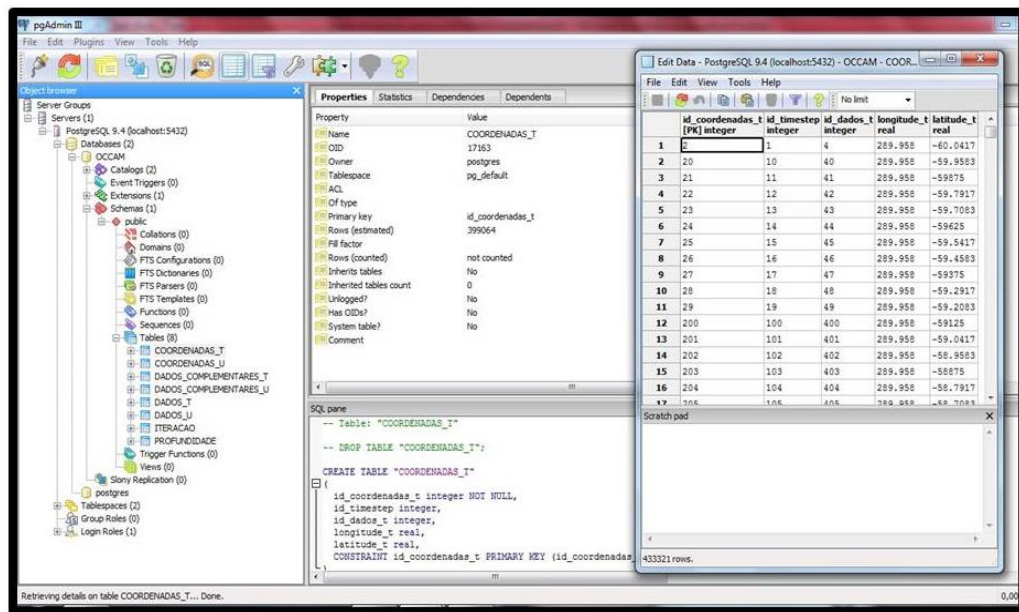


Figura 4.27: Janela para visualização dos dados armazenados na tabela COORDENADAS_T

E por fim, a última etapa foi como exportar os dados de cada variável no MATLAB para ser inseridos em cada campo correspondente de cada tabela. O processo para isso foi, após a visualização de todas as variáveis do OCCAM no MATLAB, o *Workspace* do programa listar as matrizes contendo os dados de cada variável em sua janela de comando. E, em seguida, visualizar cada matriz e copiar todos os dados, inserindo-os no campo correspondente a tabela pela qual este pertence. Para um melhor entendimento, será explicada em maiores detalhes como esta etapa foi realizada, utilizando o mesmo exemplo, da tabela COORDENADAS_T.

Como já foi visto na Figura 4.25, na tabela COORDENADAS_T, as variáveis presentes são LONGITUDE_T e LATITUDE_T. Logo, no editor do MATLAB foram executados os seguintes comandos:

```
LONGITUDE_T=ncread('01_01_88__0001.nc','LONGITUDE')
LATITUDE_T=ncread('01_01_88__0001.nc','LATITUDE')
```


A partir daí, foram geradas as suas matrizes, respectivamente, de dimensões 721x1 e 601x1 no *Workspace*, e do próprio *Workspace* os dados eram copiados e armazenados na planilha (Figura 4.26). Assim, a planilha estaria completa com todos os dados, tanto os do campo `LONGITUDE_T` e `LATITUDE_T`, como também os *ids* correspondentes a cada tabela. E como também já foi visto anteriormente, a tabela `COORDENADAS_T` possui 433.321 *ids*, isso significa que esta tabela terá 433.321 linhas geradas para os dados. Mas como chegar a este número de linhas, já que a quantidade de dados das duas variáveis dessa tabela é de respectivamente 721 elementos da `LONGITUDE_T` e 601 da `LATITUDE_T`? Para que isso seja resolvido, foram inseridos os dados dessas duas variáveis da seguinte forma: na coluna `LATITUDE_T` da tabela `COORDENADAS_T` foram inseridos todos os 601 elementos copiados do *Workspace* do MATLAB da variável e colados na coluna da planilha, repetindo esse processo 721 vezes. Depois disso, para completar a coluna `LONGITUDE_T` também foram copiados do *Workspace* dessa variável os 721 elementos, e para cada elemento copiado, foi sendo armazenado esse elemento, repetindo-o 601 vezes na coluna da planilha. Isso foi feito até o último elemento da `LONGITUDE_T` ter sido armazenado na tabela. Logo, se conclui que cada elemento da variável `LONGITUDE_T` foi combinado aos 601 elementos da variável `LATITUDE_T`, gerado 433.321 combinações. Outras tabelas com variáveis de dimensões 721x601, ao serem armazenadas não precisarão referenciar as longitude e latitudes novamente, pois uma vez que as combinações delas já foram feitas nas tabelas das `COORDENADAS` (T e U) não precisará mais armazená-las em outras tabelas, bastando armazenar o *id* (chave estrangeira) correspondente a esta combinação nas outras tabelas que possuem variáveis que geram dados a partir das combinações de longitude e latitude. E como o banco de dados é relacional, uma vez a variável declarada no banco, ela não precisará mais ser declarada em outra tabela.

Todo esse processo dividido em três etapas foi realizado para seis tabelas das oito presentes no banco de dados. As tabelas `DADOS_COMPLEMENTARES_T` e `DADOS_COMPLEMENTARES_U` foram as únicas que não pode armazenar os seus dados no *PostgreSQL*, já que houve um problema na visualização das variáveis que pertencem às tabelas. Como a primeira tabela é formada pelas variáveis

SALINITY_MEAN e POTENTIAL_TEMPERATURE_MEAN, e a segunda tabela pelas variáveis V_VELOCITY_MEAN e U_VELOCITY_MEAN, e essas quatro variáveis são de dimensões 721x601x66, não foi possível visualizá-las no *Workspace* do MATLAB, pois como já foi visto na Figura 4.22, variáveis com valores acima de 524.148 elementos não podem ser visualizados no *Workspace*. Sendo assim, as duas tabelas com essas variáveis não poderão ser armazenadas para este trabalho. Novos métodos serão pesquisados no futuro.

4.6 Análise dos Dados Armazenados

Como o objetivo principal deste trabalho é de comparar o tamanho dos arquivos com dados oceanográficos no formato *NetCDF* com o do banco de dados do *PostgreSQL*. Após armazenar todos os dados das seis tabelas no *PostgreSQL*, foi possível gerar um relatório com relação ao tamanho de armazenamento dos dados de cada tabela. O mesmo pode ser observado nas Figuras 4.28 e 4.29. O tamanho do arquivo compactado no formato *NetCDF* da rotina de coleta de dados do dia 1 de janeiro de 1988 é de 432 MB, e o tamanho gerado no banco de dados relacional no *PostgreSQL* do mesmo dia de coleta é de aproximadamente 222 MB (51,38% do tamanho do arquivo compactado no formato *NetCDF*).

Essa condição se consolida, visto que os dados armazenados no formato *NetCDF* se tornam extensos, porque há muita repetição nos dados das variáveis. Todas as variáveis de dimensões 721x601 e 721x601x66 dependem unicamente das combinações entre longitudes e latitudes (T e U) e há também algumas variáveis que dependem da profundidade para gerarem os dados. E isso se trata de muita informação de variáveis repetidas e replicadas, as quais estão armazenadas no arquivo *NetCDF*.

E pode-se considerar que a redução, quando da inclusão de mais dias de coleta do OCCAM no banco de dados, será maior, visto que as duas tabelas maiores do banco (Coordenadas_T e Coordenadas_U) são inseridas apenas uma vez, pois representam a localização das coletas. Ou seja, essas duas tabelas não serão mais alteradas.

Tables statistics report

Generated: 17/03/2016 12:25:39
 Server: PostgreSQL 9.4 (localhost:5432)
 Database: OCCAM
 Schema: public

Tables statistics

Table Name	Tuples inserted	Tuples updated	Tuples deleted	Tuples HOT updated	Live tuples	Dead tuples	Last vacuum	Last autovacuum	Last analyze	Last autoanalyze	Vac
COORDENADAS_T	0	0	0	0	0	0					0
COORDENADAS_U	0	0	0	0	0	0					0
DADOS_COMPLEMENTARES_T	0	0	0	0	0	0					0
DADOS_COMPLEMENTARES_U	0	0	0	0	0	0					0
DADOS_T	0	0	0	0	0	0					0
DADOS_U	0	0	0	0	0	0					0
ITERACAO	0	0	0	0	0	0					0
PROFUNDIDADE	0	0	0	0	0	0					0

Report generated by pgAdmin III

Figura 4.28: Relatório do PostgreSQL do tamanho de armazenamento do banco de dados relacional

Tables statistics report

Generated: 17/03/2016 12:37:36
 Server: PostgreSQL 9.4 (localhost:5432)
 Database: OCCAM
 Schema: public

Tables statistics

Table Name	Tuples inserted	Tuples updated	Tuples deleted	Tuples HOT updated	Live tuples	Dead tuples	Last vacuum	Last autovacuum	Last analyze	Last autoanalyze	Vacuum counter	Autovacuum counter	Analyze counter	Autoanalyze counter	Size
COORDENADAS_T	0	0	0	0	0	0					0	0	0	0	50 MB
COORDENADAS_U	0	0	0	0	0	0					0	0	0	0	51 MB
DADOS_COMPLEMENTARES_T	0	0	0	0	0	0					0	0	0	0	24 kB
DADOS_COMPLEMENTARES_U	0	0	0	0	0	0					0	0	0	0	8192 bytes
DADOS_T	0	0	0	0	0	0					0	0	0	0	41 MB
DADOS_U	0	0	0	0	0	0					0	0	0	0	41 MB
ITERACAO	0	0	0	0	0	0					0	0	0	0	51 MB
PROFUNDIDADE	0	0	0	0	0	0					0	0	0	0	28 MB

Report generated by pgAdmin III

Tables statistics

Table Name	Tuples inserted	Tuples updated	Last autoanalyze counter	Size
COORDENADAS_T	0	0		50 MB
COORDENADAS_U	0	0		51 MB
DADOS_COMPLEMENTARES_T	0	0		24 kB
DADOS_COMPLEMENTARES_U	0	0		8192 bytes
DADOS_T	0	0		41 MB
DADOS_U	0	0		41 MB
ITERACAO	0	0		51 MB
PROFUNDIDADE	0	0		28 MB

Figura 4.29: Relatório do PostgreSQL de tamanho de armazenamento de cada tabela no banco de dados relacional

Como já foi exposto anteriormente, há duas tabelas que não foram inseridas no banco de dados: DADOS_COMPLEMENTARES_T e DADOS_COMPLEMENTARES_U. Com isso, não se sabe qual seria o tamanho exato das mesmas, mas estima-se que ambas teriam um tamanho muito pequeno, pois armazenam apenas duas variáveis novas cada tabela, do tipo real. Esse cálculo é baseado no fato de que as tabelas que contêm as coordenadas T e U, que comprometem o maior tamanho no banco de dados do que as outras, já estão armazenadas em outras tabelas, não sendo necessário repeti-las.

5. CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho teve como objetivos a construção de um BD para os dados oceanográficos do Modelo OCCAM com base na técnica de Modelagem Entidade-Relacionamento (MER) e a realização de análises quantitativas em comparação ao tamanho de armazenamento, de um mesmo dia de coleta dos dados oceanográficos, no banco de dados relacional com o tamanho de armazenamento que atualmente os dados se encontram.

Para isso, foi realizado um estudo baseado no entendimento dessa técnica, a criação do modelo ER e sua implementação no SGBD *PostgreSQL*, a visualização e extração dos dados no *MATLAB* e a sua inserção no banco relacional.

Após essas etapas serem alcançadas, realizou-se uma análise quantitativa do tamanho do armazenamento gerado neste banco. Embora duas tabelas (com duas variáveis armazenadas em cada uma), não tenham sido incluídas, o tamanho de armazenamento do banco de dados relacional foi de aproximadamente 222 MB (Aproximadamente 50% do tamanho de armazenamento do original).

Como já foi exposto, o tamanho de armazenamento de um dia de coleta compactados no formato *NetCDF* é de 432 MB. E ainda, ao longo dos 16 anos de coletas totalizaram 1.152 coletas, exigindo um grande tamanho para o seu armazenamento durante todos esses anos de coleta. O tamanho de armazenamento no banco de dados relacional é significativamente menor e representa um ganho de armazenamento total bastante grande. Contudo, ainda vale lembrar que o banco de dados relacional de um dia de coleta dos dados oceanográficos possui uma diferença de aproximadamente 210 MB no tamanho de armazenamento, comparado ao mesmo dia de armazenamento do arquivo no formato *NetCDF*.

Além da questão de tamanho de armazenamento, os arquivos no formato *NetCDF*, a cada uso, devem ser descompactados e dados repetidos tratados para seu uso. Já os dados no banco de dados relacional, uma vez inseridos nas tabelas estão prontos para sua utilização.

Além disso, as duas maiores tabelas do banco (Coordenadas_T e Coordenadas_U) são inseridas apenas uma vez para todas as coletas, pois representam a localização das mesmas. Ou seja, essas duas tabelas não serão mais alteradas.

Este trabalho surgiu da necessidade de organizar esses dados do Modelo OCCAM em um banco de dados no Laboratório de Oceanografia da FURG. Optou-se pelo banco de dados relacional, pois os dados tratam-se de rotinas de coletas com as mesmas variáveis ao longo desses 16 anos. Sendo assim, uma vez construído o modelo ER para essas variáveis e a sua implementação, não será necessário realizar nenhum tipo de construção na sua modelagem, apenas exportar cada dia para o banco. Este trabalho trouxe como contribuição para o Laboratório, a primeiro momento, a estruturação do banco de dados relacional para os dados oceanográficos do Modelo OCCAM e a sua implementação no SGBD *PostgreSQL*. Com isso, o pesquisador da área poderá contar com outro mecanismo de busca rápido e organizado e que não exige um grande tamanho para o armazenamento dos dados, facilitando assim o trabalho no Laboratório.

Contudo, ao longo da execução desse trabalho, muitos problemas surgiram, como o de não conseguir visualizar e inserir os dados de duas tabelas do BD, o que necessitará de mais estudos para resolvê-lo.

E ainda têm-se como interesse, futuramente trabalhos de pesquisa sobre como importar todos os dados dos 16 anos de coleta e desenvolver uma interface gráfica para sua consulta com métricas de análises.

Por fim, esse trabalho teve como contribuição científica inicial a apresentação do trabalho intitulado “Estruturação de dados oceanográficos utilizando um modelo entidade-relacionamento: um estudo de caso com o modelo OCCAM”, na 14ª Mostra da Produção Universitária da FURG (2015).

REFERÊNCIAS

AKHIR, M. F. M. Surface circulation and temperature distribution of Southern South China Sea from Global Ocean Model (OCCAM). *Sains Malaysiana*, v. 41, n. 6, p. 701-714, 2012.

CHAPMAN. R.; DURRANI, T. S.; WILLEY, T. Design strategies for implementing systolic and wavefront arrays using OCCAM. *IEEE*, V. 10, p. 292-295, 1985.

CIRANO, M.; MATA, M. M.; CAMPOS, E. J. D.; DEIRÓ, N. F. R. A circulação oceânica de larga-escala na região oeste do Atlântico Sul com base no Modelo de Circulação Global OCCAM. *Revista Brasileira de Geofísica*, v. 24, n. 2, São Paulo, Apr/junho 2006.

DATE, C. J. Introdução a sistemas de banco de dados / C. J. Date; tradução de Daniel Vieira. – Rio de Janeiro: Elsevier, 2003. – 865 p.: il.; tab. – Tradução da 8ª edição americana.

ELMASRI, Ramez. Sistemas de banco de dados / Ramez Esmasri e Shamkant B. Navathe; revisor técnico Luis Ricardo de Figueiredo. – São Paulo: Pearson Addison Wesley, 2005. - 724 p.

GARRISON, Tom. Fundamentos de oceanografia / Tom Garrison; tradução técnica Cintia Miyaji...[et al]. – São Paulo: Cengage Learning, 2010. – 426 p.; il.

GONÇALVES, João. Definição de chave primária e chave estrangeira. Disponível em: <<http://joaogoncalves010.blogspot.com.br/2011/05/definicao-de-chave-primaria-e-chave.html>>. Acesso em: 30 de abril de 2015.

HEMISFÉRIO. In Britannica Escola Online. Enciclopédia Escolar Britannica, 2015. Web, 2015. Disponível em: <<http://escola.britannica.com.br/article/481476/hemisferio>>. Acesso em: 11 de março de 2015.

HEUSER, Carlos Alberto. Projeto de banco de dados. 2ª edição – Porto Alegre: Sagra Luzzatto, 1999.

HP Huang. A quick guide on how to use Matlab netCDF functions. 2009. Disponível em: <http://www.public.asu.edu/~hhuang38/matlab_netcdf_guide.pdf>. Acesso em: 13 de outubro de 2015.

LAMOURIE, Matthew J. Atlantic ocean. In Britannica Escola Online. Enciclopédia Escolar Britannica, 2015. Web, 2015. Disponível em: <<http://global.britannica.com/EBchecked/topic/41191/Atlantic-Ocean>>. Acesso em: 26 de abril de 2015.

MAFALDA, Paulo Jr. História da Oceanografia Biológica. Disponível em: <<http://www2.unifap.br/alexandresantiago/files/2012/03/1.1-Introducao-a-Oceanografia-Biologica1.pdf>>2012. Acesso em: 22 de abril de 2015.

MARTINS, Agnaldo Silva. Introdução ao estudo dos oceanos – aula 1. – Departamento de Ecologia e Recursos Naturais da Universidade Federal do Espírito Santo, Brasil. Disponível em: <<http://www.dern.ufes.br/agnaldo/oceanos/>>. Acesso em: 03 de março de 2015.

MATLAB. In Wikipédia. Enciclopédia livre. Disponível em: <<https://pt.wikipedia.org/wiki/MATLAB>>. Acesso em 12 de março de 2016.

MINSTER, Jean-Francois. A máquina oceano / Jean-Francois Minster. – Lisboa: Instituto Piaget, 1997. – 305 p.

OLIVEIRA, José Valente de. Dicionário de dados. In Web, 2000. Disponível em: <<http://w3.ualg.pt/~jvo/ep/dd.pdf>>. Acesso em 20 de maio de 2015.

O QUE é Latitude e Longitude? In: Laboratório de Cartografia – UFSM. Disponível em: <http://coral.ufsm.br/cartografia/index.php?option=com_content&view=article&id=43&Itemid=39>. Acesso em: 05 de março de 2015.

PORTAL, E. Oceanografia física, química e biológica. In: Portal – Educação. Disponível em: <<http://www.portaleducacao.com.br/biologia/artigos/7086/oceanografia-fisica-quimica-e-biologica#ixzz3YpEtZFsx>>2009. Acesso em: 03 de março de 2015.

POSTGRESQL. In Wikipedia. Enciclopédia livre. Disponível em: <<http://pt.wikipedia.org/wiki/PostgreSQL>>. Acesso em: 30 de maio de 2015.

POSTGRESQL. In PostgreSQL. Disponível em: <<http://www.postgresql.org/download/>>. Acesso em: 20 de maio de 2015.

RODRIGUES, Joel. Modelo Entidade Relacionamento (MER) e Diagrama Entidade-Relacionamento (DER). In Devmedia. Disponível em: <<http://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332>>. Acesso em 23 de maio de 2015.

SANCHES, Andre Rodrigo. Disciplina: Fundamentos de Armazenamento e Manipulação de Dados. Disponível em: <<http://www.ime.usp.br/~andrrs/aulas/bd2005-1/aula7.html>>. Acesso em: 22 de abril de 2015.

SCHIMIEGELOW, João Marcos Miragaia. O planeta azul: uma introdução as ciências marinhas / João Marcos Miragaia Schimiegelow. – Rio de Janeiro: Interciência, 2004. – 202 p.

SKINNER, Brian J. O homem e o oceano/Brian J. Skinner e Karl K. Turekian; tradução de Kenitiro Suguio. – São Paulo: E Blucher: EDUSP, 1977. – 157 p.

Drijfhout, S.S; J, Donners; V.P.M, de Ruijter The origin of Intermediate and Subpolar Mode Waters crossing the Atlantic equator in OCCAM. Geophysical Research Letters, v. 32, 2005.

TONINI, Adriana Maria; SCHETTINO, Daniela Naufel. MATLAB para engenharia. Centro Universitário de Belo Horizonte, 1ª Edição. Belo Horizonte, MG. 2002.

Anexos

Anexo 1. Scripts das oito tabelas criadas no SGBD PostgreSQL

Tabela ITERAÇÃO:

```
-- Table: "ITERACAO"
-- DROP TABLE "ITERACAO";
CREATE TABLE "ITERACAO"
(
  id_timestep integer NOT NULL,
  id_coordenadas_t integer,
  id_coordenadas_u integer,
  day real,
  years real,
  seconds real,
  CONSTRAINT id_timestep PRIMARY KEY (id_timestep),
  CONSTRAINT fk_coordenadas_t FOREIGN KEY (id_coordenadas_t)
    REFERENCES "COORDENADAS_T" (id_coordenadas_t) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT fk_coordenadas_u FOREIGN KEY (id_coordenadas_u)
    REFERENCES "COORDENADAS_U" (id_coordenadas_u) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE "ITERACAO"
```

```

OWNER TO postgres;

-- Index: fki_pk_iteraca
-- DROP INDEX fki_pk_iteraca;
CREATE INDEX fki_pk_iteraca
ON "ITERACAO"
USING btree
(id_coordenadas_t);

-- Index: fki_pk_iteracao_u
-- DROP INDEX fki_pk_iteracao_u;
CREATE INDEX fki_pk_iteracao_u
ON "ITERACAO"
USING btree
(id_coordenadas_u);

```

Tabela COORDENADAS T:

```

-- Table: "COORDENADAS_T"
-- DROP TABLE "COORDENADAS_T";
CREATE TABLE "COORDENADAS_T"
(
id_coordenadas_t integer NOT NULL,
id_timestep integer,
id_dados_t integer,
longitude_t real,
latitude_t real,
CONSTRAINT id_coordenadas_t PRIMARY KEY (id_coordenadas_t),
CONSTRAINT fk_id_dados_t FOREIGN KEY (id_dados_t)

```

```
REFERENCES "DADOS_T" (id_dados_t) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fk_id_timestep FOREIGN KEY (id_timestep)
REFERENCES "ITERACAO" (id_timestep) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

```
WITH (
```

```
  OIDS=FALSE
```

```
);
```

```
ALTER TABLE "COORDENADAS_T"
```

```
  OWNER TO postgres;
```

```
-- Index: fki_id_dados_t
```

```
-- DROP INDEX fki_id_dados_t;
```

```
CREATE INDEX fki_id_dados_t
```

```
  ON "COORDENADAS_T"
```

```
  USING btree
```

```
  (id_dados_t);
```

```
-- Index: fki_id_timestep
```

```
-- DROP INDEX fki_id_timestep;
```

```
CREATE INDEX fki_id_timestep
```

```
  ON "COORDENADAS_T"
```

```
  USING btree
```

```
  (id_timestep);
```

Tabela COORDENADAS U:

```
-- Table: "COORDENADAS_U"
```

```

-- DROP TABLE "COORDENADAS_U";
CREATE TABLE "COORDENADAS_U"
(
  id_coordenadas_u integer NOT NULL,
  id_timestep integer,
  id_dados_u integer,
  longitude_u real,
  latitude_u real,
  CONSTRAINT id_coordenadas_u PRIMARY KEY (id_coordenadas_u),
  CONSTRAINT fk_id_dados_u FOREIGN KEY (id_dados_u)
    REFERENCES "DADOS_U" (id_dados_u) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT fk_timestep FOREIGN KEY (id_timestep)
    REFERENCES "ITERACAO" (id_timestep) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE "COORDENADAS_U"
  OWNER TO postgres;
-- Index: fki_id_dados_u
-- DROP INDEX fki_id_dados_u;
CREATE INDEX fki_id_dados_u
  ON "COORDENADAS_U"
  USING btree

```

```

(id_dados_u);
-- Index: fki_timestep
-- DROP INDEX fki_timestep;
CREATE INDEX fki_timestep
ON "COORDENADAS_U"
USING btree
(id_timestep);

```

TABELA DADOS T:

```

- Table: "DADOS_T"
-- DROP TABLE "DADOS_T";
CREATE TABLE "DADOS_T"
(
    id_dados_t integer NOT NULL,
    id_coordenadas_t integer,
    id_dados_complementares_u integer,
    id_dados_complementares_t integer,
    km_t real,
    sea_surface_height_mean real,
    runoff_mean real,
    max_mixed_layer_depth_mean real,
    min_mixed_layer_depth_mean real,
    mixed_layer_depth_mean real,
    total_freshwater_into_ocean_m real,
    CONSTRAINT id_dados_t PRIMARY KEY (id_dados_t),
    CONSTRAINT fk_coordenadas_t FOREIGN KEY (id_coordenadas_t)

```

```

REFERENCES "COORDENADAS_T" (id_coordenadas_t) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT          fk_dado_complementares_t          FOREIGN          KEY
(id_dados_complementares_t)
REFERENCES          "DADOS_COMPLEMENTARES_T"
(id_dados_complementares_t) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT          fk_dado_complementares_u          FOREIGN          KEY
(id_dados_complementares_u)
REFERENCES          "DADOS_COMPLEMENTARES_U"
(id_dados_complementares_u) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE "DADOS_T"
OWNER TO postgres;

```

TABELA DADOS U:

```

-- Table: "DADOS_U"
-- DROP TABLE "DADOS_U";
CREATE TABLE "DADOS_U"
(
id_dados_u integer NOT NULL,
id_coordenadas_u integer,
id_dados_complementares_u integer,

```

```

u_wind_stress_mean real,
v_wind_stress_mean real,
CONSTRAINT id_dados_u PRIMARY KEY (id_dados_u),
CONSTRAINT fk_coordenadas_u FOREIGN KEY (id_coordenadas_u)
    REFERENCES "COORDENADAS_U" (id_coordenadas_u) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT      fk_dados_complementares_u      FOREIGN      KEY
(id_dados_complementares_u)
    REFERENCES          "DADOS_COMPLEMENTARES_U"
(id_dados_complementares_u) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE "DADOS_U"
    OWNER TO postgres;
-- Index: fki_coordenadas_u
-- DROP INDEX fki_coordenadas_u;
CREATE INDEX fki_coordenadas_u
    ON "DADOS_U"
    USING btree
    (id_coordenadas_u);

```

TABELA DADOS COMPLEMENTARES U:

```
-- Table: "DADOS_COMPLEMENTARES_U"
```



```

-- DROP TABLE "DADOS_COMPLEMENTARES_U";
CREATE TABLE "DADOS_COMPLEMENTARES_U"
(
  id_dados_complementares_u integer NOT NULL,
  id_dados_u integer,
  id_dados_t integer,
  id_depth integer,
  v_velocity_mean real,
  u_velocity_mean real,
  CONSTRAINT          id_dados_complementares_u          PRIMARY          KEY
(id_dados_complementares_u),
CONSTRAINT fk_dados_u FOREIGN KEY (id_dados_u)
  REFERENCES "DADOS_U" (id_dados_u) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fk_profundidade FOREIGN KEY (id_depth)
  REFERENCES "PROFUNDIDADE" (id_depth) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fk_dados_t FOREIGN KEY (id_dados_t)
  REFERENCES "DADOS_T" (id_dados_t) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE "DADOS_COMPLEMENTARES_U"
  OWNER TO postgres;

```

TABELA DADOS COMPLEMENTARES T:

```
-- Table: "DADOS_COMPLEMENTARES_T"
-- DROP TABLE "DADOS_COMPLEMENTARES_T";
CREATE TABLE "DADOS_COMPLEMENTARES_T"
(
  id_dados_complementares_t integer NOT NULL,
  id_dados_t integer,
  id_depth integer,
  salinity_mean real,
  potential_temperature_mean real,
  CONSTRAINT          id_dados_complementares_t          PRIMARY          KEY
(id_dados_complementares_t),
  CONSTRAINT fk_dados_t FOREIGN KEY (id_dados_t)
  REFERENCES "DADOS_T" (id_dados_t) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT fk_depth FOREIGN KEY (id_depth)
  REFERENCES "PROFUNDIDADE" (id_DEPTH) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE "DADOS_COMPLEMENTARES_T"
  OWNER TO postgres;
```

PROFUNDIDADE:

```
-- Table: "PROFUNDIDADE"
-- DROP TABLE "PROFUNDIDADE";
CREATE TABLE "PROFUNDIDADE"
(
  id_depth integer NOT NULL,
  id_complementares_u integer,
  id_complementares_t integer,
  depth real,
  CONSTRAINT id_depth PRIMARY KEY (id_depth),
  CONSTRAINT          fk_dados_complementares_u          FOREIGN          KEY
(id_dados_complementares_u)
          REFERENCES          "DADOS_COMPLEMENTARES_U"
(id_dados_complementares_u)
  MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT fk_depth FOREIGN KEY (id_depth)
          REFERENCES "PROFUNDIDADE" (id_depth) MATCH SIMPLE
          ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE "PROFUNDIDADE"
  OWNER TO postgres;
```

Anexo 2. Rotina dos comandos que foram utilizados para a visualização dos dados no MATLAB

```
% teste dados occam

% Abrir arquivos occam

ncdisp('C:\Users\Jessica Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc')

% ncdisp contem as informações essenciais sobre as variáveis e suas dimensões do
arquivo%

infoFicheiroNetCDF=ncinfo('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc')

ncid1=netcdf.open('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc')

ncid1=netcdf.open('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc','NC_NOWRITE')

% Este comando "abre" um arquivo netcdf chamado 'C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc'

% E atribui um número de arquivo para isto

% Este número de arquivo é a única saída da função e é colocado em nossa variável,
ncid1, para usos posteriores

% A opção 'NC_NOWRITE' designa que o arquivo é somente leitura

% Passo 2: Inspeccionar o número de variáveis, o número de dimensões, etc no ficheiro

[ndim,NVAR,natt,unlim]=netcdf.inq(ncid1)

% Depois de abrir o arquivo o que conter nele??? Netcdf.inq é o conj de quatro
elementos que contem o numero da dimensão, número de variáveis, número de atributos

% Neste caso, contém (NVAR=24) 24 variáveis e no máximo de 7 dimensões (ndim=7)
e 5 número de atributos (natt=5)

% Passo 3: extrair mais informações para cada dimensão

[dimname,dimlength]=netcdf.inqDim(ncid1,0)
```

```

% A primeira dimensão é o TIMESTEP com 1 elemento
[dimname,dimlength]=netcdf.inqDim(ncid1,1)
% Segunda dimensão é LONGITUDE _T com 721
[dimname,dimlength]=netcdf.inqDim(ncid1,2)
% Terceira dimensão é LATITUDE_T com 601 elementos
[dimname,dimlength]=netcdf.inqDim(ncid1,3)
% Quarta dimensão é LONGITUDE_U com 721 elementos
[dimname,dimlength]=netcdf.inqDim(ncid1,4)
% Quinta dimensão é LATITUDE_U com 601 elementos
[dimname,dimlength]=netcdf.inqDim(ncid1,5)
% Sexta dimensão é DEPTH com 66 elementos
[dimname,dimlength]=netcdf.inqDim(ncid1,6)
% Sétima dimensão é DEPTH_EDGES com 67 elementos
% Passo 4: Extração de mais informações para cada variável
[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,0)
% Extração do nome, dimensão, etc da primeira variável
% A função netcdf.inqVar(ncid1,varid), o parâmetro começa a partir de 0
% A primeira variável é TIMESTEP. É do tipo 4 (xtype=4)inteiro. dimit =0 ver o que acontece???
[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,1)
% A segunda variável é DAY do tipo 5 real ou float%
[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,2)
% A terceira variável é YEARS do tipo real ou float
[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,3)
% A quarta variável é SECONDS do tipo real ou float%

```

% A variável TIMESTEP (dimid=0) é unidimensional, isto é, matriz com sua cota definida pelo número de dimensão zero,

% Ou seja, a primeira cota ou TIMESTEP com 1 elemento. TIMESTEP é tanto uma variável e uma dimensão (ou coordenada)

[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,4)

% A quinta variável é LONGITUDE_T do tipo real ou float

% dimensão numero 1

[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,5)

% A quinta variável é LATITUDE_T do tipo real ou float

% dimensão número 2 com 601 elementos

[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,6)

% A sexta variável é KMT do tipo inteiro

[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,7)

% A sétima variável é LONGITUDE_U do tipo real ou float%

% dimensão numero 3%

[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,8)

% A oitava variável é LATITUDE_U do tipo real ou float

% dimensão numero 4

[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,9)

% A nona variável é KMU do tipo inteiro

[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,10)

% A décima variável é U_WIND_STRESS_MEAN_ do tipo real ou float

[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,11)

% A décima primeira variável é V_WIND_STRESS_MEAN_ do tipo real ou float

[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,12)

% A décima segunda é RUNOFF_MEAN_ do tipo real ou float

```

[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,13)
% A décima terceira é TOTAL_FRESHWATER_INTO_OCEAN_M do tipo real ou float
[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,14)
% A décima quarta variável é SEA_SURFACE_HEIGHT_MEAN_ do tipo real ou float
[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,15)
% A décima quinta variável é MIXED_LAYER_DEPTH_MEAN_ do tipo real ou float
[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,16)
% A décima sexta variável é MAX_MIXED_LAYER_DEPTH_MEAN é do tipo real ou
float
[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,17)
% A décima sétima variável é MIN_MIXED_LAYER_DEPTH_MEAN do tipo real ou
float
[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,18)
% A décima oitava variável é DEPTH do tipo real ou float
[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,19)
% A décima nona variável é POTENTIAL_TEMPERATURE_MEAN_ é do tipo real ou
float
[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,20)
% A vigésima variável é SALINITY_MEAN_ do tipo real ou float
% Matriz tridimensional%
[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,21)
% A vigésima primeira é U_VELOCITY_MEAN_ do tipo real ou float
[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,22)
% A vigésima segunda variável é V_VELOCITY_MEAN_ do tipo real ou float
[varname,xtype,dimid,natt]=netcdf.inqVar(ncid1,23)
% A vigésima terceira variável é DEPTH_EDGES do tipo real ou float%

```

```
LONGITUDE_T=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc',LONGITUDE_T')

LATITUDE_T=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc',LATITUDE_T')

TIMESTEP=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc','TIMESTEP')

LONGITUDE_U=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc',LONGITUDE_U')

LATITUDE_U=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc',LATITUDE_U')

DEPTH=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc','DEPTH')

DEPTH_EDGES=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc','DEPTH_EDGES')

DAY=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc','DAY')

YEARS=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc','YEARS')

SECONDS=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc','SECONDS')

KMT=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc','KMT')

KMU=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc','KMU')

U_WIND_STRESS__MEAN_=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc','U_WIND_STRESS__MEAN_')
```


V_WIND_STRESS__MEAN_=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc','V_WIND_STRESS__MEAN_')
RUNOFF__MEAN_=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc','RUNOFF__MEAN_')
TOTAL_FRESHWATER_INTO_OCEAN__M=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc','TOTAL_FRESHWATER_INTO
_OCEAN__M')
SEA_SURFACE_HEIGHT__MEAN_=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc','SEA_SURFACE_HEIGHT__M
EAN_')
MIXED_LAYER_DEPTH__MEAN_=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc','MIXED_LAYER_DEPTH__M
EAN_')
MAX_MIXED_LAYER_DEPTH__MEAN_=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc','MAX_MIXED_LAYER_DEPT
H__MEAN_')
MIN_MIXED_LAYER_DEPTH__MEAN_=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc','MIN_MIXED_LAYER_DEPTH
__MEAN_')
POTENTIAL_TEMPERATURE__MEAN_=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc','POTENTIAL_TEMPERATURE
__MEAN_')
SALINITY__MEAN_=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc','SALINITY__MEAN_')
U_VELOCITY__MEAN_=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc','U_VELOCITY__MEAN_')
V_VELOCITY__MEAN_=ncread('C:\Users\Jessica
Brum\Desktop\Arquivos_Jessica\01_01_88__0001.nc','V_VELOCITY__MEAN_')