



UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
PROGRAMA DE PÓS GRADUAÇÃO EM FÍSICA  
UNIDADE ACADÊMICA DE FÍSICA

Alcimar Araújo de Medeiros

# **Testando a Isotropia Estatística do Universo usando dados de ondas de rádio**

Campina Grande, Paraíba, Brasil

19 de março de 2024

Alcimar Araújo de Medeiros

# **Testando a Isotropia Estatística do Universo usando dados de ondas de rádio**

Dissertação realizada sob orientação do Prof. Dr. Amilcar Rabelo de Queiroz, apresentada à Unidade Acadêmica de Física em complementação aos requisitos para obtenção do título de Mestre em Física.

Orientador: Professor Dr. Amilcar Rabelo de Queiroz

Campina Grande, Paraíba, Brasil

19 de março de 2024



MINISTÉRIO DA EDUCAÇÃO  
**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE**  
POS-GRADUACAO EM FISICA

Rua Aprigio Veloso, 882, - Bairro Universitario, Campina Grande/PB, CEP 58429-900

**FOLHA DE ASSINATURA PARA TESES E DISSERTAÇÕES**

**ALCIMAR ARAÚJO DE  
MEDEIROS**

Testando a Isotropia Estatística do Universo, usando dados de ondas de rádio

Dissertação apresentada ao Programa de Pós-Graduação em Física como pré-requisito para obtenção do título de Mestre em Física.

Aprovada em: 06/09/2023

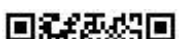
**Amílcar Rabelo de Queiroz**  
Presidente da Comissão e Orientador

**João Rafael Lúcio dos Santos**  
Examinador Interno

**Allysson Macário de Araújo Caldas**     
Examinador Externo



Documento assinado eletronicamente por **AMILCAR RABELO DE QUEIROZ, PROFESSOR 3 GRAU**, em 06/09/2023, às 19:13, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da [Portaria SEI nº 002, de 25 de outubro de 2018](#).





A autenticidade deste documento pode ser conferida no site <https://sei.ufcg.edu.br/autenticidade>, informando o código verificador **3772821** e o código CRC **5E1A08CB**.

---

**Referência:** Processo nº 23096.069051/2023-34

SEI nº 3772821

M488t      Medeiros, Alcimar Araujo de.  
              Testando a isotropia estatística do universo usando dados de ondas  
              de rádio / Alcimar Araujo de Medeiros. – Campina Grande, 2024.  
              34 f. : il. color.

              Dissertação (Mestrado em Física) – Universidade Federal de  
Campina Grande, Centro de Ciências e Tecnologia, 2023.  
              "Orientação: Prof. Dr. Amilcar Rabelo de Queiroz".  
              Referências.

              1. Cosmologia. 2. Isotropia. 3. Galaxias. 4. Universo. 5. Ondas de  
Rádio. I. Queiroz, Amilcar Rabelo de. II. Título.

CDU 524.8(043)

*A Deus, pela minha vida, e por me permitir ultrapassar todos os obstáculos encontrados ao longo da realização deste trabalho.*

# Agradecimentos

- A Deus, por ter permitido que eu tivesse saúde e determinação para não desanimar durante a realização deste trabalho.
- Ao Prof. Dr. Amilcar Rabelo de Queiroz, pela orientação, sugestão, estímulo e competência com os qual conduziu este trabalho até aqui.
- A todos os professores do curso de Mestrado em Física que contribuíram com a minha formação.
- Aos meus familiares, pelo apoio, confiança e incentivo.
- Aos meus amigos professores da ECI Maria Lídia Rangel, Tenório - PB.
- Aos colegas de pós-graduação e funcionários da Unidade Acadêmica de Física, pela grata convivência durante a minha permanência no curso.
- A todos que direto ou indiretamente possibilitaram a realização e conclusão deste trabalho.

*“Não vos amoldeis às estruturas deste mundo,  
mas transformai-vos pela renovação da mente,  
a fim de distinguir qual é a vontade de Deus:  
o que é bom, o que Lhe é agradável, o que é perfeito.  
(Bíblia Sagrada, Romanos 12, 2)*



# Resumo

Sabemos que a cosmologia estuda o cosmo, ou seja, o Universo desde sua origem, estrutura e evolução. Com o passar dos anos, novas descobertas vêm sendo feitas, como por exemplo a de Hubble, em que o pesquisador observou que as galáxias estavam ficando cada vez mais distantes, e afastada uma das outras. [1]. O objetivo do nosso trabalho é encontrar a Isotropia Estatística do Universo, usando dados de ondas de rádio. Levando isso em consideração, baixamos um catálogo (NVSS), através do Site Vizier [2] para analisamos o número de galáxias do céu. Em consequência, tivemos condições de mostrar uma visão geral do Catálogo que foi utilizado para obter os dados e todos os métodos, incluindo os procedimentos que utilizamos para baixar a tabela para o computador e poderemos descrevermos as bibliotecas do python utilizadas para encontramos o número de galáxias. Por fim, construímos os gráficos empregando o matplotlib em Coordenadas Equatoriais e Galácticas, como também, o healpy. Depois disso, aplicamos uma máscara no mapa, aplicando, desse modo, na tabela (A máscara é uma forma de filtrar os dados, mantendo apenas os objetos que atendem a certas condições, enquanto os objetos que não satisfazem essas condições são excluídos ou ignorados). Por conseguinte, descrevemos o percurso que seguimos no Jupyter notebook desde a criação das bibliotecas até os gráficos para caso alguém venha reproduzir esses mapas.

**Palavras-chave:** Cosmologia. Isotropia. Galaxias. Universo. Ondas de Rádio.

# Abstract

We know that cosmology studies the cosmos, that is, the universe from its origin, structure, and evolution. Over the years, new discoveries have been made, and one of these discoveries was made by Hubble, in which the researcher observed that galaxies were becoming increasingly distant from each other and, at the same time, were moving away more quickly [1]. The aim of our work is to find the Statistical Isotropy of the Universe, using radio wave data. In view of this, we downloaded a catalog (NVSS) through the Vizier Site [2] to analyze the number of galaxies in the sky. As a result, we were able to show an overview of the Catalog that was used to obtain the data, and all the methods, including the procedures we used to download the table to the computer and we were able to describe the python libraries that we used to find the number of galaxies. Finally, we constructed the graphs using matplotlib in Equatorial and Galactic Coordinates, as well as healpy. Following that, we applied a mask (The mask is a way to filter data, keeping only the objects that meet certain conditions, while objects that do not satisfy these conditions are excluded or ignored) to the map, and then applied it to the table. As a result, we outlined the path we followed in the Jupyter notebook, from library creation to graph generation, in case someone wishes to reproduce these maps.

**Keywords:** Cosmology. Isotropy. Galaxies. Universe. Radio Waves.

# Lista de ilustrações

Figura 1 – Healpix . . . . .	14
Figura 2 – Resumo do número de pixels e seus tamanhos acessíveis ao HEALPix. [3]	15
Figura 3 – The Very Large Array, New México . . . . .	18

# Lista de abreviaturas e siglas

HEALPix	Hierarchical Equal Area Iso Latitude Pixelation of a sphere
$\Lambda$ CDM	Lambda-Cold Dark Matter.
NVSS	NRAO VLA SkY SURVEY
CMB	Radiação C3smica de fundo em micro-ondas
VLA	Very Large Array

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
<b>1.1</b>	<b>Expansão do Universo</b>	<b>11</b>
<b>1.2</b>	<b>Equação de Friedmann</b>	<b>12</b>
<b>2</b>	<b>PESQUISAS, TESTES E ANÁLISE</b>	<b>14</b>
<b>2.1</b>	<b>Introdução</b>	<b>14</b>
2.1.1	Software HEALPix	14
2.1.2	Coordenadas Celestes	15
<b>2.2</b>	<b>Dados observacionais</b>	<b>17</b>
2.2.1	O Very Large Array ( VLA)	17
2.2.2	O NRAO VLA SkY SURVEY (NVSS)	18
<b>2.3</b>	<b>Seleção de Amostra</b>	<b>19</b>
<b>3</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>22</b>
<b>3.1</b>	<b>Jupyter Notebook</b>	<b>22</b>
<b>4</b>	<b>CONCLUSÃO</b>	<b>33</b>
	<b>REFERÊNCIAS</b>	<b>34</b>

# 1 Introdução

Atualmente, sabemos que a Cosmologia estuda o Cosmo, ou seja, o Universo desde sua origem, estrutura e evolução [1]. No entanto, desde a antiguidade, muitos pesquisadores já estudavam essa área de pesquisa. Antes do século XIX, era impossível observamos o Universo além da nossa galáxia, somente depois do Século XIX, com o uso de novas tecnologias, conseguimos visualizar o cosmo, além da nossa galáxia. O astrônomo Edwin Hubble teve muito esforço para medir as distâncias das galáxias. Em 1929, observou-se que as galáxias estavam ficando cada vez mais distantes umas das outras, como também, mais depressa se afastavam [1] [4].

Após muitas medidas, Hubble constatou que a velocidade com a qual as galáxias se afastavam era proporcional a distância que as separava umas das outras [5]. Portanto, concluiu que o Universo estava em expansão, introduzindo assim um novo conceito de modelo de Universo. Se o Universo está se expandindo, então, ele era menor no passado e deve ter tido uma origem a partir de um estado infinitamente quente e denso [4].

Através do princípio cosmológico, conseguimos estudar o Universo, onde o mesmo afirma que o Universo é homogêneo e isotrópico [1]. Os dois princípios juntos implicam que o Universo é uniforme [6]. O Universo é homogêneo e isotrópico em escalas espaciais suficientemente grandes ( $> 100$  Mpc) [1]. Essa homogeneidade afirma que em escalas muito grandes, o Universo tem uma densidade média praticamente semelhante em todos os pontos. Já o termo isotrópico implica dizer que em todos os pontos do Universos temos a mesma aparência [1].

O princípio cosmológico claramente não é válido em escalas pequenas, ou seja, somente em escalas muito grandes. Logo, o Universo parece realmente uniforme e o princípio cosmológico é válido [1]. O evento que deu origem ao Universo é chamado Big Bang. A cosmologia adota como modelo padrão para explicar o início do Universo, com base na teoria do Big Bang pelo qual explica que o Universo primitivo se expandiu de um estado inicialmente quente e denso para seu estado atual relativamente frio que continua em expansão até os dias de hoje [7].

A Cosmologia moderna surgiu da Teoria Geral da Relatividade de Einstein. Como a gravidade era a única força de alcance infinito que poderia agir sobre a matéria neutra, Einstein percebeu que o Universo como um todo deve obedecer às suas leis [4]. Ele foi levado a acreditar que o Universo era estático ou imutável com o tempo, causando, dessa maneira, um problema real. Com a gravidade sendo uma força atrativa, naturalmente, faria com que objetos estacionários no espaço desmorona-se até um ponto [4].

Para superar isso, Einstein teve que introduzir um termo em sua equações chamada

de constante cosmológica  $\lambda$  ( $\lambda$ ). Isto representa uma forma de antigravidade que tem a propriedade interessante de que seus efeitos se tornam maior com a distância [4]. Então, com uma força diminuindo e a segunda aumentando com a distância, foi possível produzir uma solução estática. Mais tarde, Einstein percebeu que isso era uma situação instável e que um Universo estático não era possível, chamando isso de o maior erro de sua vida [4].

Assim, Einstein poderia ter previsto que o Universo estaria expandindo ou contraindo. No entanto, como veremos, talvez ele não estava tão errado como pensava [4]. Um dos modelos usado pelos cosmólogos para explicar o Big Bag, é o modelo  $\lambda$ CDM. Esse Lambda, nada mais é do que a constante cosmológica que está associada a energia escura do Universo. Através do Big Bang, o físico americano George Gamow percebeu que deveria ter uma radiação que ainda permeava o Universo até os dias de hoje [4]. No entanto, como se sabe, a radiação descoberta em 1965 por A.A. Penzias e R. W. Wilson [1], foi uma descoberta por acaso. Essa radiação passou a ser chamada de Radiação Cós mica de Fundo em Micro-ondas (CMB).

Sabemos que o principio Cosmológico garante que Universo é homogêneo e isotrópico. No entanto, ainda encontramos poucos testes nessa área de pesquisa. Com isso, nosso trabalho consiste testar essa isotropia do Universo, usando os dados de ondas de rádio através do Catalogo, NVSS. Para isso, utilizamos os dados do NVSS para fazer as simulações do céu, mostrando essa isotropia.

## 1.1 Expansão do Universo

Uma peça principal em Cosmologia de observação é que quase tudo no Universo parece se afastar de nós, e quanto mais distante algo estar, mais rápida sua recessão parece ser [1]. Essas velocidades são medidas através do desvio para o vermelho, que é basicamente o efeito Doppler aplicado às ondas de luz [8]. As galáxias têm um conjunto de absorção e linhas de emissão identificáveis em seus espectros, cujas frequências características são bem conhecidas. No entanto, se uma galáxia está se movendo em nossa direção, as ondas de luz se aglomeram aumentando a frequência [1]. Isso ocorre porque a luz azul está na extremidade de alta frequência da especificação visível, sendo tal fenômeno denominado blueshift [4]. Se a galáxia está se recuando, as linhas características se movem para a extremidade vermelha do espectro, gerando um efeito conhecido como redshift [1].

Edwin Hubble percebeu que com esse afastamento, as galáxias se afastavam uma da outra devido aos desvios para o vermelho. Em consequência disso, ajustou seus dados com a equação da relação linear, que hoje conhecemos como a lei de Hubble [8]. Acontece que quase todas as galáxias estão se afastando de nós, a maioria apresentava deslocamento espectral redshift ( $Z$ ) para o vermelho [8]. Esse redshift ( $Z$ ) é definido por:

$$Z = \frac{\lambda_{\text{obs}} - \lambda_{\text{em}}}{\lambda_{\text{em}}} \quad (1.1)$$

Onde  $\lambda_{\text{obs}}$  e  $\lambda_{\text{em}}$  são os comprimentos de onda da luz nos pontos de emissão das galáxias. Logo, se uma galáxia próxima está se afastando a uma velocidade  $v$ , então, seu redshift ( $Z$ ) é dado por:

$$Z = v/c \quad (1.2)$$

Onde  $c$  é a velocidade da luz.

Interpretando os desvios para o vermelho como desvios Doppler, Hubble mostrou que a velocidade da recessão  $V$  era proporcional a distância  $r$ . Logo, escreveu que:

$$V = H_0 r \quad (1.3)$$

Isso é conhecido como Lei de Hubble e a Constante de proporcionalidade  $H_0$  como Constante de Hubble. A Lei de Hubble não é exata, pois o princípio cosmológico não é válido perfeitamente para galáxias próximas, que normalmente possuem alguns movimentos aleatórios conhecidos como velocidades peculiares [8]. Mas descreve o comportamento médio das galáxias muito bem [8].

## 1.2 Equação de Friedmann

A equação de Friedmann descreve a expansão do Universo e, portanto, é uma das equações mais importantes em cosmologia. Tal equação nos permite trabalhar resolvendo sob diferentes suposições sobre o Universo [1]. Essa equação foi formulada pelo matemático e físico russo-americano Alexander Friedmann em 1922, baseando-se nas equações de campo da teoria da relatividade geral de Albert Einstein. Em essência, a equação de Friedmann relaciona a taxa de expansão do Universo, representada pela Constante de Hubble ( $H$ ), com a densidade de matéria e energia presentes no Universo, a constante cosmológica ( $\Lambda$ ) e a curvatura espacial ( $K$ ) [8].

A equação geral da equação de Friedmann é dada por:

$$H^2 = \frac{8\pi G}{3}\rho - \frac{k}{a^2} + \frac{\Lambda}{3} \quad (1.4)$$

Onde:

- $H$  é a Constante de Hubble, que mede a taxa de expansão do Universo;
- $G$  é a constante gravitacional de Newton;



- $\rho$  é a densidade total de matéria e energia no Universo;
- $k$  é a constante cosmológica, relacionada à curvatura espacial do Universo;
- $a$  é o fator de escala, que representa o tamanho relativo do Universo em relação a um momento de referência.

A equação de Friedmann permite estudar a evolução do Universo ao longo do tempo, fornecendo informações essenciais sobre seu destino. Existem três cenários possíveis para a geometria do Universo, dependendo do valor da curvatura espacial ( $K$ ):

- Universo Aberto ( $k < 0$ ): Curvatura é negativa, de modo que nesse tipo de Universo, a taxa de expansão diminui ao longo do tempo, mas nunca chega a parar e a expansão do Universo continua indefinidamente. Esse cenário sugere um destino de resfriamento cósmico contínuo.

- Universo Fechado ( $k > 0$ ): Neste caso, a curvatura é positiva e a expansão do Universo eventualmente desacelerará e se reverterá, levando a um grande colapso. Essa situação implica que o Universo tem uma quantidade finita de matéria e energia e, eventualmente, de forma que o levar para uma singularidade gravitacional.

- Universo Plano ( $k = 0$ ): Aqui, a curvatura é nula, e a expansão do Universo continuará indefinidamente, mas com uma taxa de desaceleração gradual. Esse cenário é consistente com um Universo infinito e em expansão constante.

Em suma, a equação de Friedmann nos permite entender como o Universo em larga escala está evoluindo e como suas propriedades fundamentais, como a densidade de matéria e energia, a constante cosmológica e a curvatura, afetam sua expansão ao longo do tempo.

## 2 Pesquisas, testes e Análise

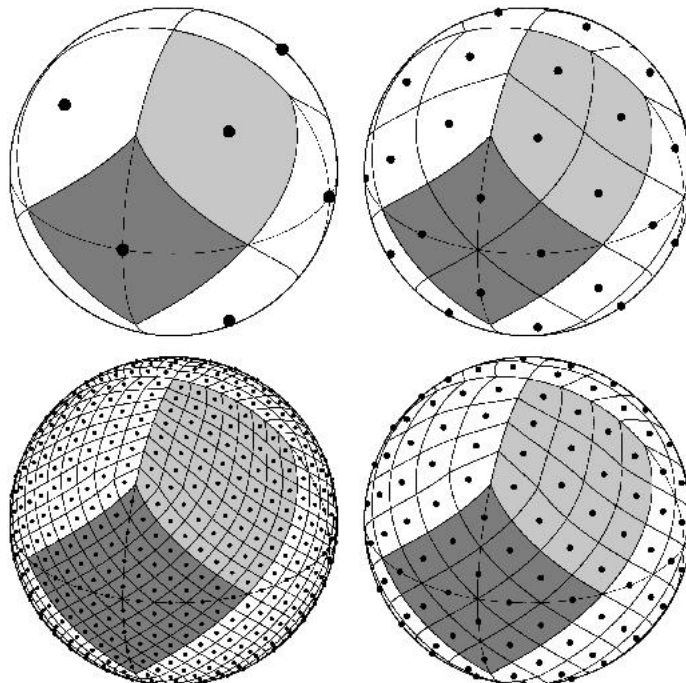
### 2.1 Introdução

Neste capítulo, vamos mostrar uma visão geral do catálogo que foi usado para obter os dados e os métodos, incluindo os procedimentos que utilizamos para baixar a tabela para o computador, além de descrever as bibliotecas do python que empregamos no desejo de encontrar o número de galáxias.

#### 2.1.1 Software HEALPix

HEALPix é um acrônimo para "Hierarchical Equal Area isoLatitude Pixelization of a sphere". Como sugerido no nome, essa pixelização produz uma subdivisão de uma superfície esférica na qual cada pixels cobre a mesma área de superfície que todos os outros pixels [3]. O HEALPix atende a esses requisitos porque possui as três propriedades essenciais a seguir:

Figura 1 – Healpix



Fonte: <https://healpix.sourceforge.io/>

1. A esfera é hierarquicamente tessellada em quadriláteros curvilíneos. A partição de resolução mais baixa é composta por 12 pixel de base. A resolução da tesselação aumenta pela divisão de cada pixel em quatro novos. O aumento da resolução em três

etapas ocorre a partir do nível base (ou seja, a esfera é particionada, respectivamente, em 12, 48, 192 e 768 pixels).

2. As áreas de todos os pixels em uma determinada resolução são idênticas.

3. Os pixels são distribuídos em linhas de latitude constante. Esta propriedade é essencial para todas as aplicações de análise harmônica, envolvendo harmônicos esféricos. A área de cada pixel é definida como:

$$\Omega_{pix} = \pi/3N^2_{side} \quad (2.1)$$

Figura 2 – Resumo do número de pixels e seus tamanhos acessíveis ao HEALPix. [3]

$k$	$N_{side} = 2^k$	$N_{pix} = 12N_{side}^2$	$\theta_{pix} = \Omega_{pix}^{1/2}$
0	1	12	58.6°
1	2	48	29.3°
2	4	192	14.7°
3	8	768	7.33°
4	16	3072	3.66°
5	32	12288	1.83°
6	64	49152	55.0'
7	128	196608	27.5'
8	256	786432	13.7'
9	512	3145728	6.87'
10	1024	12582912	3.44'
11	2048	50331648	1.72'
12	4096	201326592	51.5''
13	8192	805306368	25.8''
14	$2^{14}$	$3.22 \times 10^9$	12.9''
15	$2^{15}$	$1.29 \times 10^{10}$	6.44''
16	$2^{16}$	$5.15 \times 10^{10}$	3.22''
17	$2^{17}$	$2.06 \times 10^{11}$	1.61''
⋮	⋮	⋮	⋮
29	$2^{29}$	$3.46 \times 10^{18}$	$3.93 \times 10^{-4}''$

Fonte: <https://iopscience.iop.org/article/10.1086/427976/pdf>

### 2.1.2 Coordenadas Celestes

As **coordenadas celestes** Compõem um sistema usado pelos astrônomos para localizar objetos no céu. Este sistema é baseado na posição dos objetos celestes em relação à Terra [9]. Existem duas coordenadas celestes principais: a ascensão reta e a declinação. A ascensão reta é medida em horas, minutos e segundos e é análoga à longitude na Terra. A ascensão reta é medida em horas porque a Terra gira em torno de seu eixo uma vez a cada 24 horas. A declinação é medida em graus e é análoga à latitude na Terra. Ela mede o ângulo norte-sul de um objeto celeste a partir do equador celeste, que é uma projeção da linha do equador terrestre na esfera celeste.

A declinação pode variar de  $-90^\circ$  a  $+90^\circ$  com valores positivos, indicando objetos celestes ao norte do equador celeste e valores negativos indicando objetos celestes ao sul. Juntos, a ascensão reta e a declinação fornecem as coordenadas celestes precisas para localizar qualquer objeto no céu. Ressaltamos que existem vários tipos de coordenadas celestes, cada um com uma finalidade específica. Alguns dos principais tipos de coordenadas celestes são:

As **coordenadas equatoriais** são um sistema de coordenadas celestes que se pautam no equador celeste. Esse sistema de coordenadas é amplamente utilizado para localizar objetos no céu e descrever sua posição [9]. Existem duas principais coordenadas equatoriais: a ascensão reta e a declinação. A ascensão reta é medida em horas, minutos e segundos de arco é medida a partir do ponto em que o equador celeste cruza a linha do horizonte ao leste.

A declinação é medida em graus, minutos e segundos de arco é medida a partir do equador celeste. A ascensão reta é análoga à longitude na Terra, enquanto a declinação é análoga à latitude. As coordenadas equatoriais são usadas para descrever a posição de objetos no céu em relação a um observador na Terra. Uma das vantagens das coordenadas equatoriais é que elas são independentes do local de observação. Isso significa que a mesma posição no céu será descrita pelas mesmas coordenadas equatoriais, independentemente, de onde o observador esteja na Terra. No entanto, as coordenadas equatoriais mudam ao longo do tempo devido ao movimento da Terra em torno do Sol.

As **coordenadas horizontais** são um sistema de coordenadas celestes que é baseado no horizonte local do observador. Esse sistema de coordenadas é útil para descrever a posição de objetos no céu em relação à localização do observador [9]. As coordenadas horizontais são dadas em duas medidas: altitude e azimute. A altitude é medida em graus acima do horizonte, enquanto o azimute é medido em graus a partir do norte, no sentido horário. Juntos, esses valores fornecem uma descrição completa da posição de um objeto no céu em relação ao observador. Ao contrário das coordenadas equatoriais, as coordenadas horizontais não são fixas no tempo e mudam constantemente à medida que a Terra gira em torno de seu eixo. Para converter as coordenadas equatoriais em coordenadas horizontais, é necessário conhecer a latitude e a longitude do local de observação.

As **coordenadas eclípticas** são um sistema de coordenadas celestes que é baseado no plano da eclíptica, que é o plano orbital da Terra em torno do Sol. Esse sistema de coordenadas é útil para descrever a posição de objetos no céu em relação à eclíptica [9]. As coordenadas eclípticas são dadas em duas medidas: longitude eclíptica e latitude eclíptica. A longitude eclíptica é medida em graus por meio do ponto de vernal, que é o ponto onde o equador celeste cruza a eclíptica no equinócio de primavera. A latitude eclíptica é medida em graus mediante a eclíptica em direção ao norte ou ao sul.

As coordenadas eclípticas são particularmente úteis para descrever a posição dos planetas e outros objetos do sistema solar em relação à Terra. Ao contrário das coordenadas equatoriais, as coordenadas eclípticas mudam lentamente ao longo do tempo devido ao movimento da Terra em torno do Sol. Para converter as coordenadas equatoriais em coordenadas eclípticas, é necessário levar em conta a inclinação da eclíptica em relação ao equador celeste.

As **coordenadas galácticas** são um sistema de coordenadas celestes baseado no plano da Via Láctea, a galáxia em espiral à qual pertence o nosso sistema solar. Esse sistema de coordenadas é útil para descrever a posição de objetos no céu em relação à estrutura da galáxia [9]. As coordenadas galácticas são dadas em duas medidas: longitude galáctica e latitude galáctica. A longitude galáctica é medida em graus a partir do centro galáctico, enquanto a latitude galáctica é medida em graus através do plano galáctico em direção ao norte ou ao sul. Ao contrário das coordenadas equatoriais e eclípticas, as coordenadas galácticas mudam lentamente ao longo do tempo devido ao movimento da Terra em torno do centro da galáxia. Para converter as coordenadas equatoriais em coordenadas galácticas, é necessário levar em conta a inclinação da galáxia em relação ao equador celeste.

## 2.2 Dados observacionais

### 2.2.1 O Very Large Array ( VLA)

O Very Large Array (VLA) é um observatório de radioastronomia localizado na Planície de San Agustin, entre as cidades de Magdalena e Dátil, a aproximadamente 80 km a oeste de Socorro, no Novo México. É o radiotelescópio mais versátil e amplamente utilizado no mundo. Ele pode mapear estruturas em larga escala de gás e nuvens moleculares, da mesma maneira que identificar ejeções de plasma de buracos negros supermassivos. É a primeira câmera colorida do mundo para radioastronomia, graças ao seu novo conjunto de receptores e um supercomputador que pode processar amplos campos de dados espectrais simultaneamente [10].

O VLA consiste em 27 antenas independentes, cada um com um disco de diâmetro de 25 metros e pesando 209 toneladas. As antenas são ordenadas ao longo de três braços, formando um Y (cada braço mede 21 km). Eles não estão todos organizados em uma linha, como mostra a figura 3. A resolução angular que pode ser alcançada, está entre 0,2 e 0,04 segundos de arco. Existem quatro configurações comumente usadas, designadas de A (a maior) a D (a mais apertada, quando todas as antenas estão a 600 metros do ponto central). A cobertura de frequência é de 74 MHz a 50 GHz [11].

Em síntese, o Very Large Array se destaca como um protagonista na exploração da

Figura 3 – The Very Large Array, New México



Fonte: [https://en.Wikipédia.org/wiki/Very Large Array](https://en.Wikipédia.org/wiki/Very_Large_Array)

radioastronomia, erguendo-se em uma localização afastada que permite captar os murmúrios cósmicos na tranquilidade do deserto. Sua flexibilidade, abrangência e capacidade de mergulhar na rica variedade das ondas de rádio que permeiam o Universo, têm sido de suma importância para nós pesquisadores, enriquecendo a nossa compreensão do cosmos. Indubitavelmente, o VLA desempenha um papel crucial em nossa incessante busca por desvendar os segredos do vasto Universo que nos cerca.

### 2.2.2 O NRAO VLA SKY SURVEY (NVSS)

O NRAO VLA Sky Survey (NVSS) é um levantamento contínuo de rádio que cobre o céu ao norte de  $-40$  graus (82% da esfera celeste) a 1,4 GHz de declinação [12]. Os principais produtos de dados do NVSS são um conjunto de 2326 "cubos" contínuos de  $4 \times 4$  graus com três planos, contendo imagens Stokes I, Q e U, além de um catálogo de quase 2 milhões de fontes discretas mais fortes do que uma densidade de fluxo de cerca de 2,5 mJy [12]. O NRAO VLA Sky Survey (1998) está disponível na base de dados Vizier [2]. Assim, usamos os dados deste catálogo, baixando, antes, as informações referentes à ascensão reta e declinação em graus, bem como a densidade de fluxo médio medida em 1400 MHz, medida em mJy.

Uma coisa realmente impressionante sobre o NVSS que ele consegue ver com muito detalhe, isto, identifica com precisão fontes individuais de rádio lá no espaço. E isso é ótimo porque nos permite estudar essas fontes em profundidade, entender como são e onde estão distribuídas, o que tem ajudado muito em áreas importantes, como entender como as galáxias evoluem ao longo do tempo [12].

O NVSS se tornou uma ferramenta supervaliosa para nós pesquisadores. Eles usam os seus dados para fazer pesquisas em grande escala, investigar o que acontece no Universo. A variedade de informações que ele fornece, juntamente com a qualidade dos dados, é realmente incrível. Por causa disso, o NVSS é como um ponto de referência

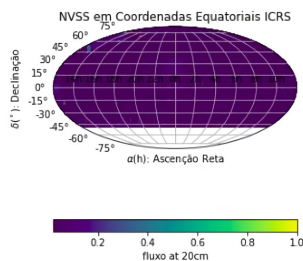
muito importante quando se trata de olhar para o céu com ondas de rádio. E, no final das contas, tudo isso nos ajuda a entender melhor o Universo, analisando as emissões de rádio que vêm de lá [12].

## 2.3 Seleção de Amostra

Fizemos o importe do catálogo NVSS, através do vizier para o jupyter notebook. Como a tabela era bem grande, baixamos apenas uma de suas partes. Para não ter problema de espera no carregamento dessa tabela, baixamos a mesma para uma pasta do computador. Feito isso, fizemos uma análise da tabela como também algumas conversões de unidades.

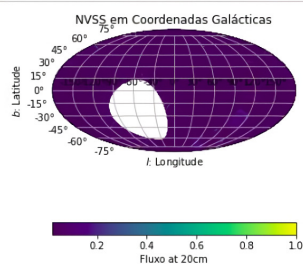
A fim de realizar um teste mais robusto da isotropia da distribuição angular, realizamos uma cuidadosa limpeza e seleção dos dados. Daí, escolhemos os seguintes limites de fluxo para nossos dados: pegamos a densidade de fluxo (S) maiores que 20 mJy e menores que 1000 mJy. Utilizando a biblioteca do matplotlib, temos os gráficos a seguir. Primeiro, o mapa em coordenadas Equatoriais:

```
In [16]: fig = plt.figure(figsize=(10, 9))
ax1 = fig.add_subplot(121, projection="mollweide")
ax1.set_title('NVSS em Coordenadas Equatoriais ICRS')
im1=ax1.scatter(x=gals_eqt.ra.wrap_at(180*u.deg).radian, y=gals_eqt.dec.radian, s = 200, c = Fluxosrenov)
ax1.set_xticklabels(['14h', '16h', '18h', '20h', '22h', '0h', '2h', '4h', '6h', '8h', '10h'])
ax1.set_xlabel(r'$\alpha$(h): Ascensão Reta')
ax1.set_ylabel(r'$\delta$(\textcircled{c}): Declinação')
ax1.grid(True)
fig.colorbar(im1, ax=ax1, label='fluxo at 20cm', orientation='horizontal');
```

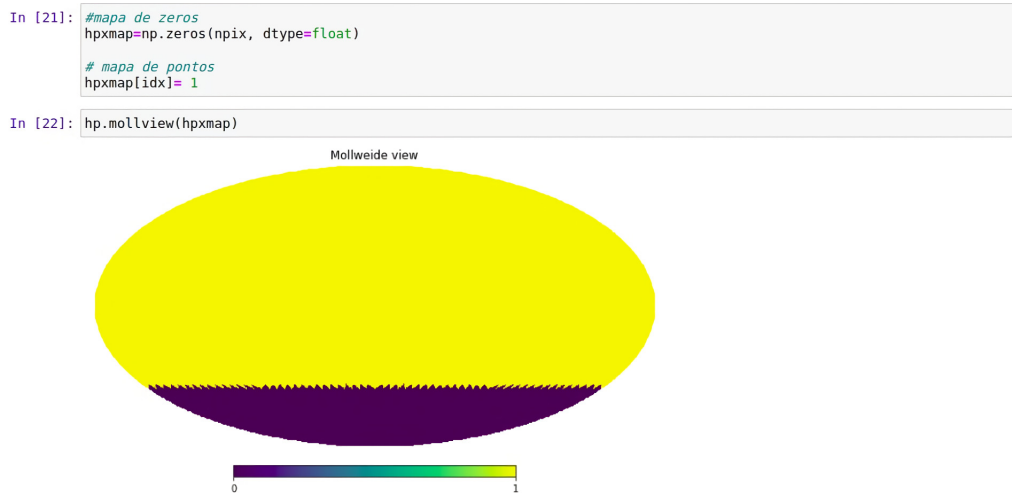


Agora, vamos fazer o mesmo gráfico em coordenadas galácticas. Logo, temos:

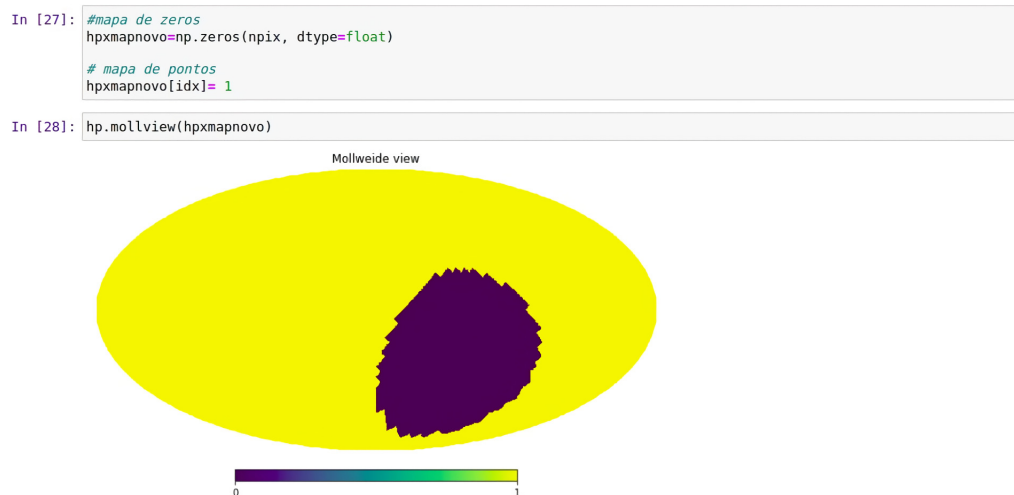
```
In [17]: fig = plt.figure(figsize=(10, 9))
ax2 = fig.add_subplot(122, projection="mollweide")
ax2.set_title('NVSS em Coordenadas Galácticas')
im2=ax2.scatter(x=gals_gal.l.wrap_at(180*u.deg).radian, y=gals_gal.b.radian, s = 200, c = Fluxosrenov)
ax2.set_xlabel(r'$l$: Longitude')
ax2.set_ylabel(r'$b$: Latitude')
ax2.grid(True)
fig.colorbar(im2, ax=ax2, label='Fluxo at 20cm', orientation='horizontal');
```



Em continuidade, fizemos os gráficos usando a biblioteca do Healpy. HEALPix (Hierarchical Equal Area isoLatitude Pixelisation) é um algoritmo para pixelizar uma esfera que, as vezes, é usado em Astronomia para armazenar dados de levantamentos de todo o céu. Como sugerido no nome, essa pixelização produz uma subdivisão de uma superfície esférica na qual cada pixel cobre a mesma área de superfície que todos os outros pixels. Começando com o gráfico em coordenadas equatoriais, temos:



Agora, temos o gráfico em coordenadas galácticas usando o Healpy.

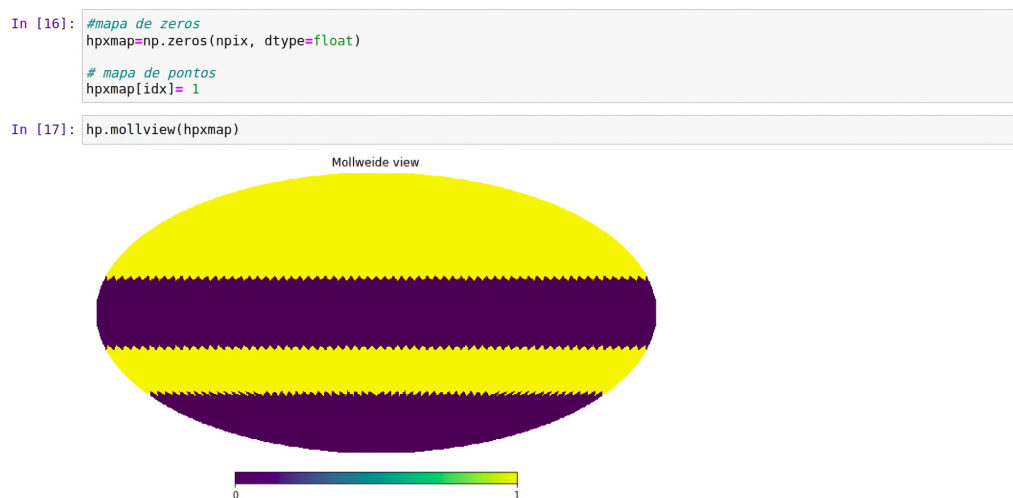


Em busca de dados mais eficazes, aplicamos uma máscara no mapa em coordenadas equatoriais do healpy com o objetivo de filtrar os dados, mantendo apenas os objetos que atendem a certas condições estabelecida por nós, enquanto os objetos que não satisfazem essas condições são excluídos. Diante disso, criamos nossos pontos de interesse que queremos excluir dessa região do mapa, ou seja, estou marcando como zero todos os pixels que estão dentro da faixa de latitude e longitude definida pela máscara conforme temos abaixo.





Agora, resolvemos aplicar a máscara na tabela, na perspectiva de que o resultado do mapa seja o mesmo, o que ocasionou no resultado seguinte:



Notamos que o resultado foi o mesmo que esperávamos, ou seja, os mapas deveriam ser iguais.

## 3 Resultados e Discussões

### 3.1 Jupyter Notebook

Agora, mostramos detalhadamente o passo a passo que empregamos nas produções dos mapas feitos no Jupyter Notebook. Primeiro, importamos as seguintes bibliotecas para o Jupyter notebook:

```
In [1]: from astroquery.vizier import Vizier
import astropy.coordinates as coord
from astropy.table import Table
from astropy.time import Time
from astropy.timeseries import TimeSeries
import astropy.units as u
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import cartopy.crs as ccrs
import numpy as np
import pandas as pd
import glob
from astroquery.simbad import Simbad
import healpy as hp
from astropy.table import Table
```

WARNING: version mismatch between CFITSIO header (v4.0009999999999999) and linked library (v4.01).

WARNING: version mismatch between CFITSIO header (v4.0009999999999999) and linked library (v4.01).

WARNING: version mismatch between CFITSIO header (v4.0009999999999999) and linked library (v4.01).

É de suma importância o uso das bibliotecas python, pois, as mesmas nos permite fazer cálculos avançados com rapidez, como também simular mapas celestes para melhor entendemos o Universo ao nosso redor. Essas bibliotecas nos ajudam a organizar informações complicadas, como coordenadas celestes ou séries temporais, de uma maneira que podemos entendê-las.

Depois de importamos as bibliotecas, utilizamos o `astroquery.vizier` para acessar catálogos astronômicos online usando o serviço Vizier para isso, pegamos nosso catálogo (NVSS) para fazemos essa pesquisa. Onde baixarmos uma tabela, como também, selecionamos as colunas que queríamos obter.

```
In [4]: NVSS_table = "VIII/65/nvss"
catalog = Vizier(catalog=NVSS_table, columns=['*', 'Full', '_Glon', '_Glat', '_RAJ2000', '_DEJ2000', 'NVSS', 'RAJ2000', 'DEJ2000'])
NVSS = catalog.query_constraints()[NVSS_table]
```

Logo após, baixamos e importamos a tabela para o Jupyter Notebook. Percebemos que a mesma é muito grande e ficava demorando um certo tempo toda vez que ia rodar a célula. Para coibir esse tal problema, resolvemos baixar essa tabela para uma pasta do computador, usando a biblioteca `pandas`. Para isso, importamos a biblioteca `pandas` objetivando fazer um Dataframe (que nada mais é do que uma estrutura de dados tabular bidimensional potencialmente heterogênea e de tamanho variável com eixos rotulados de

linhas e colunas). Agora, após fazemos o Dataframe com o pandas, criamos um nome para nosso arquivo.

```
In [5]: df_NVSS = NVSS.to_pandas()
df_NVSS.to_csv('NVSS_tabela.csv')
```

Feito isso, verificamos se o arquivo criado se encontra no diretório do computador, usando a biblioteca glob. Em Python, o módulo glob é usado para recuperar arquivos/nomes de caminho que correspondem a um padrão especificado. Este módulo vem integrado ao Python, portanto, não houve necessidade de instalá-lo externamente. Logo, usando a biblioteca, temos:

```
In [6]: files=glob.glob('/home/alcimar/*NVSS_tabela.csv')
files
Out[6]: ['/home/alcimar/NVSS_tabela.csv']
```

Notamos que o arquivo se encontra no diretório, então, abrimos e verificamos quantas linhas e colunas tem a tabela, bem como se dá a sua denominação.

```
In [7]: NVSS = pd.read_csv(files[0])
NVSS.shape
Out[7]: (1773484, 17)
```

Em continuidade, abrimos a tabela, denominando-a de "NVSS" e dá shift Enter. Percebemos que devemos reduzir essa tabela em uma quantidade menor. Para isso, antes de iniciamos, precisamos saber a quantidade mínima e máxima da densidade de fluxo de energia que se encontra abaixo da tabela.

```
In [8]: NVSS
Out[8]:
```

	Unnamed: 0	_Glon	_Glat	_RAJ2000	_DEJ2000	NVSS	RAJ2000	DEJ2000	e_RAJ2000	e_DEJ2000	S1.4	e_S1.4	l_MajAxis	M
0	0	357.481372	-76.927982	0.000375	-34.119306	000000-340709	00 00 00.09	-34 07 09.5	0.48	6.7	2.7	0.5	<	
1	1	343.289299	-74.384136	0.000542	-38.441250	000000-382628	00 00 00.13	-38 26 28.5	0.65	11.5	3.1	0.6	<	
2	2	92.681844	-63.853818	0.000667	-4.045083	000000-040242	00 00 00.16	-04 02 42.3	0.03	0.6	106.8	3.2	<	
3	3	110.448247	-29.412608	0.000750	32.209250	000000+321233	00 00 00.18	+32 12 33.3	0.31	3.9	3.7	0.4	<	
4	4	114.376920	-12.711980	0.000792	49.303500	000000+491812	00 00 00.19	+49 18 12.6	0.92	9.0	2.4	0.5	<	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1773479	1773479	10.097234	-78.074663	359.999833	-31.165444	235959-310955	23 59 59.96	-31 09 55.6	0.13	1.7	9.6	0.6	<	
1773480	1773480	103.855420	-48.351376	359.999833	12.605889	235959+123621	23 59 59.96	+12 36 21.2	0.36	5.6	2.9	0.4	<	
1773481	1773481	99.163415	-56.569814	359.999917	3.909472	235959+035434	23 59 59.98	+03 54 34.1	0.83	9.4	2.7	0.5	<	
1773482	1773482	87.239777	-67.841296	359.999958	-8.608333	235959-083630	23 59 59.99	-08 36 30.0	0.11	1.7	9.1	0.5	<	
1773483	1773483	111.470196	-25.450676	359.999958	36.273611	235959+361625	23 59 59.99	+36 16 25.0	0.16	1.9	7.2	0.5	<	

1773484 rows × 17 columns

Antes de criamos uma nova tabela, devemos analisar o número máximo e mínimo da densidade de fluxo (S) para fazemos uma filtragem desses valores. Essa filtragem é justamente para fazemos uma limpeza em nossos dados de tal forma que os mapas dos números de galaxias fiquem mais robustos. Logo, temos:

```
In [9]: print(NVSS['S1.4'].min(),NVSS['S1.4'].max())
2.0 858423.0
```

Já que conhecemos a quantidade mínima e máxima da densidade de fluxo, a fim de realizamos um teste mais robusto da isotropia da distribuição angular, foi preciso fazer uma cuidadosa limpeza e seleção dos dados. Com isso, criamos uma nova tabela, como também um nome. A partir dela filtramos a densidade de fluxo maior que 20 mJy e menor que 1000 mJy. Logo, obtivemos a seguinte tabela:

```
In [10]: Nova_tabela= NVSS[(NVSS['S1.4']>20) & (NVSS['S1.4']<1000)]
Nova_tabela
Out[10]:
```

Unnamed: 0	Glon	Glat	_RAJ2000	_DEJ2000	NVSS	RAJ2000	DEJ2000	e_RAJ2000	e_DEJ2000	S1.4	e_S1.4	l_MajAxis	M
2	92.681844	-63.853818	0.000667	-4.045083	000000-040242	00 00 00.16	-04 02 42.3	0.03	0.6	106.8	3.2	<	
6	96.877611	-59.561328	0.001625	0.683333	000000+004100	00 00 00.39	+00 41 00.0	0.03	0.6	149.5	5.3	NaN	
25	118.162040	5.772561	0.006167	68.166889	000001+681000	00 00 01.48	+68 10 00.8	0.08	0.6	220.3	6.6	<	
26	76.365614	-72.766346	0.006458	-14.728028	000001-144340	00 00 01.55	-14 43 40.9	0.05	0.8	23.0	0.8	<	
30	96.050958	-60.529763	0.006875	-0.369361	000001-002209	00 00 01.65	-00 22 09.7	0.03	0.6	212.5	6.4	<	
...	...	...	...	...	...	...	...	...	...	...	...	...	
1773454	1773454	52.395703	-77.386512	359.996958	-22.347278	23 59 59.27	-22 20 50.2	0.04	0.7	33.8	1.4	NaN	
1773458	1773458	91.049362	-65.198401	359.997250	-5.561833	23 59 59.34	-05 33 42.6	0.09	2.5	38.8	2.3	NaN	
1773460	1773460	108.540442	-36.009534	359.997375	25.420306	23 59 59.37	+25 25 13.1	0.03	0.6	89.3	2.7	<	
1773462	1773462	95.030828	-61.606608	359.997625	-1.553833	23 59 59.43	-01 33 13.8	0.03	0.6	71.5	2.2	<	
1773473	1773473	90.147389	-65.886287	359.998583	-6.344528	23 59 59.66	-06 20 40.3	0.04	0.6	47.3	1.5	<	

326290 rows × 17 columns

Agora, realizamos algumas transformações de coordenadas celestes. Pois, a declinação ( DEJ 2000) está em graus, minutos e segundos de arco. Com isso, devemos convertermos esses valores para graus. A unidade das declinações é especificada como graus quando usamos o parâmetro `unit = u.deg`. Isso nos permite trabalhar com as declinações no formato de graus. Como também devemos converter a ascensão reta (RAJ 2000) que se encontra em horas, minutos e segundos para horas. A unidade das ascensões retas é especificada como horas, usando o parâmetro `unit = u.hour`, o que nos permite trabalhar com as ascensões retas no formato de horas. Fazendo a transformação das coordenadas ICRS para coordenadas galácticas conforme a tabela, temos:

```
In [11]: ra = coord.Longitude(Nova_tabela['RAJ2000'], unit = u.hour)
dec = coord.Latitude(Nova_tabela['DEJ2000'], unit = u.deg)
gals_eqt = coord.SkyCoord(ra, dec, frame='icrs')
gals_gal = coord.SkyCoord(ra, dec, frame='icrs').transform_to('galactic')
```

Agora, vamos analisar o número mínimo da densidade de fluxo dessa nova tabela.

```
In [14]: Nova_tabela['S1.4'].min()
Out[14]: 20.1
```

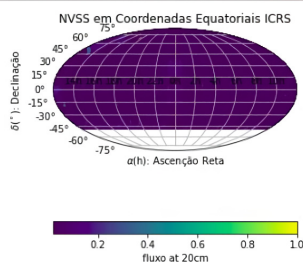
Em seguida, fizemos um filtro dessa nova tabela em que pegamos a densidade de fluxo de energia e dividimos pelo valor mínimo. Ao dividir a densidade de fluxo pelo valor mínimo, estamos normalizando os valores, logo:

```
In [15]: Fluxosrenov=Nova_tabela['S1.4']/Nova_tabela['S1.4'].max()
Fluxosrenov
Out[15]: 2      0.106832
        6      0.149545
        25     0.220366
        26     0.023007
        30     0.212564
        ...
        1773454 0.033810
        1773458 0.038812
        1773460 0.089327
        1773462 0.071521
        1773473 0.047314
Name: S1.4, Length: 326290, dtype: float64
```

Agora, vamos esboçar o gráfico usando o matplotlib (é uma biblioteca para a visualização de dados em Python, dando importância que a mesma apresenta uma API orientada a objetos que permitem a criação de gráficos em 2D de uma forma simples e com poucos comandos. A ferramenta disponibiliza diversos tipos de gráficos, como em barra, em linha, em pizza, histogramas entre outras opções), em coordenadas equatoriais (Sistema equatorial de coordenadas é um sistema de coordenadas celestes que tem como plano fundamental o Equador Celeste, sendo amplamente usado para mapear corpos celestes, tendo como coordenadas a declinação e a ascensão reta).

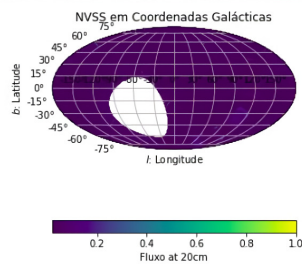
O sistema equatorial de coordenadas é muito usado já que é capaz de indicar a posição de astros suficientemente distantes da Terra de forma igual para todos os observadores, isto é, as coordenadas equatoriais (declinação e ascensão reta) de um dado astro celeste que são idênticas para qualquer observador na superfície da Terra). Logo abaixo, temos o passo a passo para o esboço do gráfico em coordenadas equatoriais, como também uma ilustração.

```
In [16]: fig = plt.figure(figsize=(10, 9))
ax1 = fig.add_subplot(121, projection="mollweide")
ax1.set_title('NVSS em Coordenadas Equatoriais ICRS')
im1=ax1.scatter(x=-gals_eqt.ra.wrap_at(180*u.deg).radian, y=gals_eqt.dec.radian, s = 200, c =Fluxosrenov)
ax1.set_xticklabels(['14h', '16h', '18h', '20h', '22h', '0h', '2h', '4h', '6h', '8h', '10h'])
ax1.set_xlabel(r'$\alpha$(h): Ascensão Reta')
ax1.set_ylabel(r'$\delta$(\textcircled{h}): Declinação')
ax1.grid(True)
fig.colorbar(im1,ax=ax1,label='fluxo at 20cm',orientation='horizontal');
```



Agora, apresentamos os mesmos gráficos em coordenadas galácticas. Logo, temos:

```
In [17]: fig = plt.figure(figsize=(10, 9))
ax2 = fig.add_subplot(122, projection='mollweide')
ax2.set_title('NVSS em Coordenadas Galácticas')
im2=ax2.scatter(x=gals_gal.l.wrap_at(180*u.deg).radian, y=gals_gal.b.radian, s = 200, c = Fluxosrenov)
ax2.set_xlabel(r'$l$: Longitude')
ax2.set_ylabel(r'$b$: Latitude')
ax2.grid(True)
fig.colorbar(im2, ax=ax2, label='Fluxo at 20cm', orientation='horizontal');
```



Por fim, esboçamos tais gráficos, usando a biblioteca do Healpy. O healpy é um pacote python para lidar com dados pixelados na esfera. De início, como queríamos esboçar o gráfico em coordenadas equatoriais, fizemos as seguintes conversões. Vamos converter a declinação para radiano.

```
In [18]: # Coordenadas Equa
ra_hp=ra.radian
#dec_hp=(np.pi/2-dec.radian)
dec_hp= np.pi/2-dec.radian
```

Feito isso, agora vamos colocar a resolução do nosso mapa. A resolução de um mapa com a biblioteca Healpy é determinada pelo parâmetro `nside`. O `nside` é um número inteiro que controla o número de pixels em cada lado do cubo que envolve a esfera. Quanto maior o valor de `nside`, maior a resolução do mapa. Logo, temos;

```
In [19]: nside=16
npix = hp.nside2npix(nside)
print(f'nside = {nside}, npix = {npix}')
nside = 16, npix = 3072
```

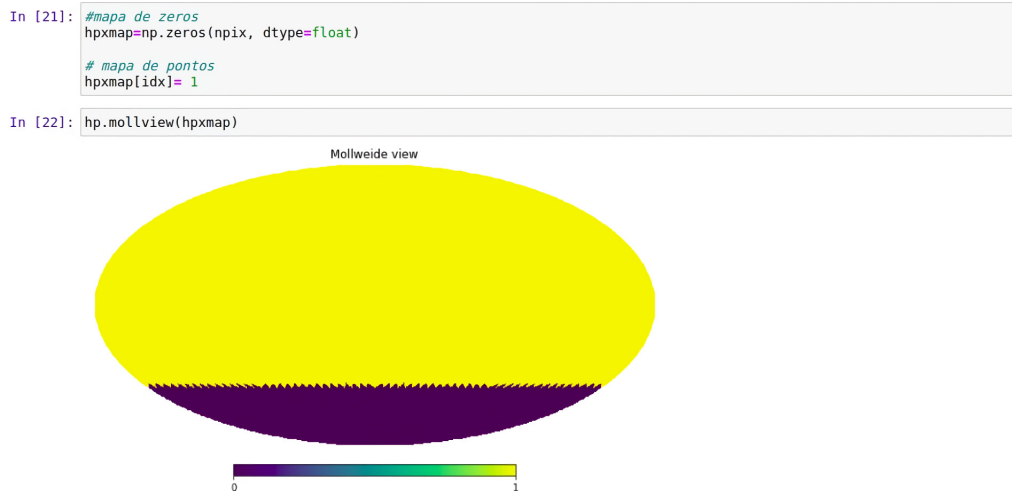
Sabemos que o parâmetro `healpix nside`, deve ser uma potência de 2 como também, devemos colocar a coordenadas angulares de um ponto na esfera. A função `healpy.pixelfunc.nside2npix` dá o número de pixels NPIX do mapa. Logo, temos:

```
In [20]: #Coordenadas Equa
idx=hp.ang2pix(nside, dec_hp, ra_hp)
idx
Out[20]: array([1568, 1440, 112, ..., 800, 1568, 1696])
```

Face a isso, pudemos fazer nosso mapa de pontos, usando o healpy em coordenadas equatoriais. Vale salientar que o sistema equatorial de coordenadas é também baseado em dois ângulos: a ascensão reta e a declinação. Uma das grandes importância dos mapas em coordenadas equatorial é o fato de permitir a visualização dos equivalentes celestes ao equador e polos geográficos.

Eles são naturalmente chamados de equador celeste e polos celestes norte e sul. O fato de haver um único plano equatorial, reconhecido como tal por todos os observadores em Terra, independente de onde estejam na sua superfície, constitui-se na condição básica

para a definição de um sistema de coordenadas que seja universal. Ou seja, utilizável por todos os observadores e cujas coordenadas para uma dada fonte que tenham valor respectivamente de quem as mede. Este é o sistema de coordenadas equatoriais. Logo, temos:



Agora, fizemos o mapa em coordenadas galácticas, usando o Healpix. No entanto, vale salientar que a posição de um astro qualquer na Esfera Celeste pode ser definido sem ambiguidade através de dois ângulos em relação ao sistema de coordenadas adotado que, por sua vez, é definido a partir de um ponto central. Para uma esfera (qualquer uma em princípio), os sistemas de referências utilizados são definidos por um plano principal que divide a esfera em duas partes iguais, definindo-se assim, um grande círculo.

Para as coordenadas galácticas, o plano principal é definido pelo plano do disco da Via Láctea (nossa galáxia é uma espiral, provavelmente barrada, com a distribuição da maioria das estrelas em um disco), o Equador Galáctico. De início, usamos o  $l$  que é um vetor de posições em coordenada  $l$  (das coordenadas galácticas), enquanto o  $b$  para a coordenada  $b$  (das coordenadas galácticas).

Em coordenadas galácticas, temos a longitude ( $l$ ) análoga à longitude em um sistema de coordenadas terrestres. No entanto, em vez de medir o ângulo em relação ao meridiano de Greenwich, é medida em relação a um plano de referência que passa pelo centro da Via Láctea e pelo ponto onde a linha do equador galáctico cruza a eclíptica (plano da órbita da Terra).

A longitude galáctica varia de  $0^\circ$  a  $360^\circ$ , medindo o ângulo no plano galáctico a partir de um ponto de referência. E a latitude ( $b$ ) é análoga ela mede o ângulo vertical acima ou abaixo do plano galáctico. A latitude galáctica varia de  $-90^\circ$  a  $+90^\circ$ , com  $0^\circ$  correspondendo ao plano galáctico propriamente dito, e valores positivos, indicando posições acima desse plano, enquanto valores negativos indicam posições abaixo dele.

Portanto, ao utilizar as coordenadas galácticas ( $l$ ,  $b$ ), estamos especificando a

posição de um ponto no céu galáctico, com a longitude galáctica indicando a direção angular ao redor da Via Láctea e a latitude galáctica a posição vertical nesse plano.

```
In [19]: # Coordenadas Galact
li=gals_gal.l
li.rad

Out[19]: array([1.6176036 , 1.69083353, 2.06231679, ..., 1.89438828, 1.65860108,
1.57336901])

In [20]: # Coordenadas Galact
bi=gals_gal.b
bi.rad

Out[20]: array([-1.11445937, -1.03954128,  0.10075021, ..., -0.62848492,
-1.07523815, -1.14993265])

In [21]: # Coordenadas Galact
li_hp=li.radian
bi_hp= np.pi/2-bi.radian
```

Sabemos que o parâmetro `nside` deve ser uma potência de 2 como também, colocamos a coordenadas angulares de um ponto na esfera. Dito de outro modo, a função `healpy.pixelfunc.nside2npix` dá o número de pixels NPIX do mapa. A projeção com Mollweide é a ferramenta de visualização mais comum para mapas HEALPix.:

```
In [21]: # Coordenadas Galact
li_hp=li.radian
bi_hp= np.pi/2-bi.radian

In [26]: # Coordenadas Galact
idx=hp.ang2pix(nside, bi_hp, li_hp)
idx
idx.shape

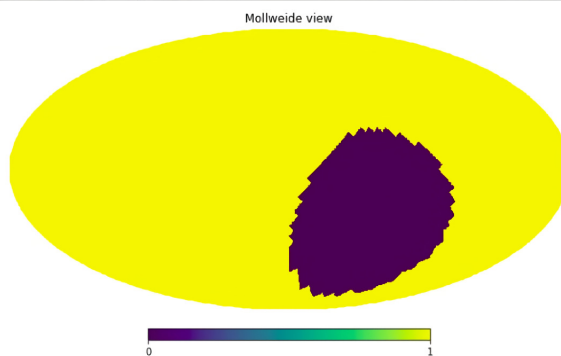
Out[26]: (326290,)
```

Assim, temos o gráfico em coordenadas galácticas usando o Healpy.

```
In [27]: #mapa de zeros
hpxmapnovo=np.zeros(npix, dtype=float)

# mapa de pontos
hpxmapnovo[idx]= 1

In [28]: hp.mollview(hpxmapnovo)
```



No presente momento, vamos aplicar uma máscara no mapa (A máscara é uma forma de filtrar os dados, mantendo apenas os objetos que atendem a certas condições, enquanto os objetos que não satisfazem essas condições são excluídos ou ignorados). Diante disso, vamos criarmos nossos pontos de interesse que queiramos excluir dessa região do mapa ou seja, que marcamos como zero todos os pixels que estão dentro da faixa de latitude e longitude definida pela máscara. Como também, converter nossa latitude ( $b$ ) e longitude ( $l$ ) de graus para radianos.

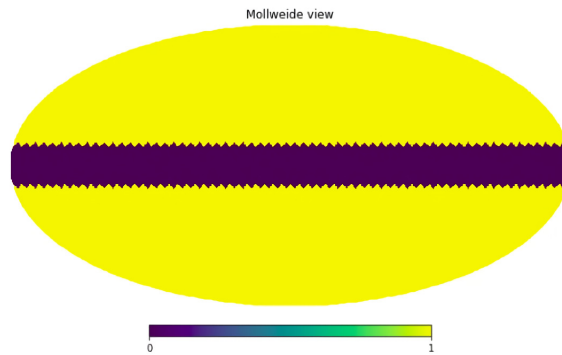


```
In [30]: b_mask= np.radians(80)
         l_mask= np.radians(100)
         idx_mask= hp.query_strip(inside, b_mask, l_mask)

In [31]: hpx_mask= np.ones(npix, dtype= float)
         hpx_mask[idx_mask]= 0
```

Aplicando a máscara, temos:

```
In [32]: hp.mollview(hpx_mask)
```

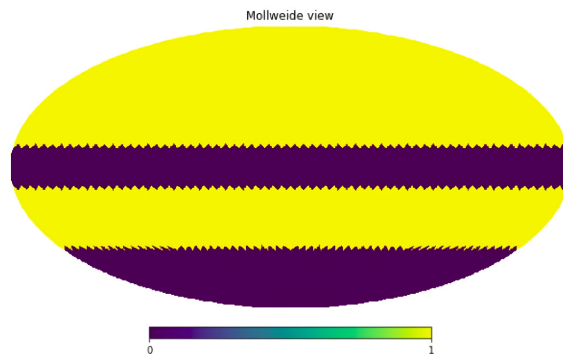


Por fim, criamos uma nova variável onde a mesma vai multiplicar o mapa que fizemos em coordenadas equatoriais vezes o mapa da máscara. Logo,

```
In [33]: NOVO_MAPA = hpxmap*hpx_mask
```

Agora, temos o mapa em coordenadas equatoriais junto com a máscara.

```
In [34]: hp.mollview(NOVO_MAPA)
```



Como aplicamos uma máscara no mapa, operamos essa máscara na tabela. Quando aplicamos uma máscara primeiro no mapa e depois na tabela, garantimos que os objetos selecionados se relacionem exatamente com a área de interesse mostrada no mapa, o que nos permitiu fazer análises mais precisas e relevantes, focando apenas nos objetos pertencentes a área que nos interessa, e excluindo aqueles que estão fora dela.

Além disso, aplicar a mesma máscara no mapa e na tabela evita erros de associação ou seja, apenas os objetos que estão dentro da área especificada pela máscara são considerados e relacionados ao mapa. Dessa forma, podemos ter mais confiança nos resultados obtidos a partir dessa seleção cuidadosa. De início, analisamos qual o mínimo

da declinação. Assim,

```
In [6]: Nova_tabela['_DEJ2000'].min()
Out[6]: -40.3736667
```

Continuando, fizemos uma mudança de variável, como também uma filtragem na declinação.

```
In [7]: NV= Nova_tabela[(Nova_tabela['_DEJ2000']<=-20) | (Nova_tabela['_DEJ2000']>=20)]
```

Agora, vamos abrir nossa nova tabela;

```
In [8]: NV
Out[8]:
```

	Unnamed: 0	_Glon	_Glat	_RAJ2000	_DEJ2000	NVSS	RAJ2000	DEJ2000	e_RAJ2000	e_DEJ2000	S1.4	e_S1.4	l_MajAxis	M
25	25	118.162040	5.772561	0.006167	68.166889	000001+681000	00 00 01.48	+68 10 00.8	0.08	0.6	220.3	6.6	<	
42	42	25.477821	-78.579656	0.009083	-28.022444	000002-280120	00 00 02.18	-28 01 20.8	0.06	0.9	21.1	0.8	<	
51	51	115.981553	-5.000354	0.009958	57.176167	000002+571034	00 00 02.39	+57 10 34.2	0.07	0.6	118.7	3.9	NaN	
53	53	112.791862	-19.945851	0.010250	41.911694	000002+415442	00 00 02.46	+41 54 42.1	0.06	0.7	33.1	1.4	NaN	
54	54	112.126343	-22.795164	0.010458	38.995778	000002+385944	00 00 02.51	+38 59 44.8	0.05	0.7	41.6	1.6	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1773418	1773418	117.937527	4.701405	359.990375	67.072667	235957+670421	23 59 57.69	+67 04 21.6	1.26	7.3	37.3	4.2	NaN	
1773433	1773433	110.030093	-30.921847	359.993708	30.657333	235958+303926	23 59 58.49	+30 39 26.4	0.04	0.6	171.6	6.2	NaN	
1773437	1773437	113.279774	-17.732271	359.994208	44.172889	235958+441022	23 59 58.61	+44 10 22.4	0.07	0.8	22.8	0.8	<	
1773454	1773454	52.395703	-77.386512	359.996958	-22.347278	235959-222050	23 59 59.27	-22 20 50.2	0.04	0.7	33.8	1.4	NaN	
1773460	1773460	108.540442	-36.009534	359.997375	25.420306	235959+252513	23 59 59.37	+25 25 13.1	0.03	0.6	89.3	2.7	<	

191489 rows × 17 columns

Novamente, fazendo uma conversão de unidades na ascensão reta onde está sendo convertida de horas (u.hour) para o formato apropriado de coordenada galácticas ( angulares) como também, a declinação sendo convertida de graus (u.deg) para coordenadas galácticas. Logo;

```
In [9]: rav = coord.Longitude(NV['_RAJ2000'], unit = u.hour)
decv = coord.Latitude(NV['_DEJ2000'], unit = u.deg)
gals_eqt1 = coord.SkyCoord(rav, decv, frame='icrs')
gals_gal1 = coord.SkyCoord(rav, decv, frame='icrs').transform_to('galactic')
```

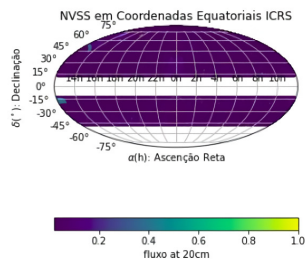
Agora, vamos dividir a densidade de fluxo da tabela pelo seu valor máximo, ao dividir a densidade de fluxo da tabela pelo seu valor máximo, estamos normalizando os valores de fluxo em relação ao maior valor encontrado na tabela.

```
In [10]: FR=NV['S1.4']/NV['S1.4'].max()
FR
Out[10]: 25      0.220366
         42      0.021106
         51      0.118736
         53      0.033110
         54      0.041612

         1773418  0.037311
         1773433  0.171651
         1773437  0.022807
         1773454  0.033810
         1773460  0.089327
Name: S1.4, Length: 191489, dtype: float64
```

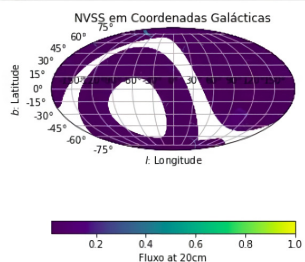
Logo, temos o gráfico em coordenadas equatoriais junto com a máscara usando o matplotlib.

```
In [11]: fig = plt.figure(figsize=(10, 9))
ax1 = fig.add_subplot(121, projection="mollweide")
ax1.set_title('NVSS em Coordenadas Equatoriais ICRS')
im1=ax1.scatter(x=gals_eqtl.ra.wrap_at(180*u.deg).radian, y=gals_eqtl.dec.radian, s = 200, c =FR)
ax1.set_xticklabels(['14h', '16h', '18h', '20h', '22h', '0h', '2h', '4h', '6h', '8h', '10h'])
ax1.set_xlabel(r'$\alpha$(h): Ascensão Reta')
ax1.set_ylabel(r'$\delta$(\textcircled{c}): Declinação')
ax1.grid(True)
fig.colorbar(im1,ax=ax1,label='fluxo at 20cm',orientation='horizontal');
```



Agora, o mapa em Coordenadas Galácticas usando também o matplotlib:

```
In [12]: fig = plt.figure(figsize=(10, 9))
ax2 = fig.add_subplot(122, projection="mollweide")
ax2.set_title('NVSS em Coordenadas Galácticas')
im2=ax2.scatter(x=gals_gall.l.wrap_at(180*u.deg).radian, y=gals_gall.b.radian, s = 200, c = FR)
ax2.set_xlabel(r'$l$: Longitude')
ax2.set_ylabel(r'$b$: Latitude')
ax2.grid(True)
fig.colorbar(im2, ax=ax2, label='Fluxo at 20cm', orientation='horizontal');
```



Fazendo novamente as seguintes conversões de unidade e colocando a resolução, temos:

```
In [13]: # Coordenadas Equa
rav_hp=rav.radian
#dec_hp=(np.pi/2-dec.radian)
decv_hp= np.pi/2-decv.radian
```

```
In [14]: nside=16
npix = hp.nside2npix(nside)
print(f'nside = {nside}, npix = {npix}')

nside = 16, npix = 3072
```

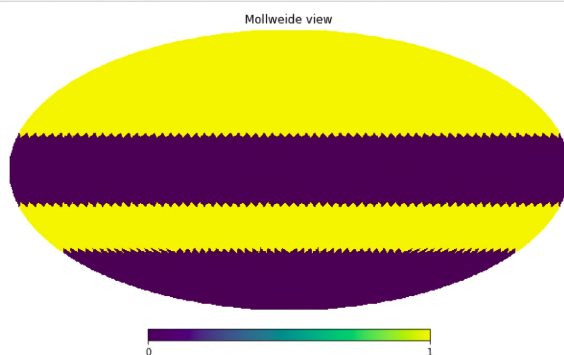
Calculando o índice de pixel (idx) com base nas coordenadas angulares, temos:

```
In [15]: #Coordenadas Equa
idx=hp.ang2pix(nside, decv_hp, rav_hp)
idx
```

Por fim, esboçamos o mapa com a máscara feito no healpy:

```
In [16]: #mapa de zeros
hpxmap=np.zeros(npix, dtype=float)
# mapa de pontos
hpxmap[idx]= 1
```

```
In [17]: hp.mollview(hpxmap)
```



## 4 Conclusão

O trabalho apresentou o estudo da cosmologia, dando destaque ao estudo do Universo através de um catálogo, investigando sua origem, estrutura e evolução onde envolveu a coleta e análise de dados através de ondas de rádio, uma ferramenta que permite a investigar do cosmo. Ao longo dos anos, a pesquisa nesse campo tem revelado diversas descobertas, incluindo a observação de Hubble, que identificou a expansão das galáxias e seu afastamento gradual umas das outras.

Nesse contexto, o foco do trabalho foi direcionado para a busca da Isotropia Estatística do Universo, utilizando dados de ondas de rádio. Para atingir esse objetivo, foi adquirido um catálogo de dados (NVSS) por meio do site Vizier [2], com o intuito de analisar o número de galáxias presentes no céu. Esse processo permitiu apresentar uma visão detalhada do catálogo utilizado, juntamente com a descrição dos métodos empregados, incluindo os procedimentos para o download dos dados e a exploração das bibliotecas Python utilizadas para a análise do número de galáxias.

A etapa subsequente envolveu a construção de gráficos, empregando as bibliotecas matplotlib para coordenadas equatoriais e galáticas, bem como o Healpy. Além disso, aplicamos uma máscara nos mapas, como também fizemos o inverso, ou seja, aplicamos a máscara na tabela. A utilização dessa máscara possibilitou a filtragem dos dados, mantendo apenas os objetos que satisfazem determinadas condições, enquanto os que não atendem a essas condições foram excluídos (ignorados).

Por fim, o percurso utilizado no Jupyter Notebook foi minuciosamente detalhado, desde a importação das bibliotecas até a geração dos gráficos, com o intuito de permitir que outros pesquisadores possam replicar os mapas gerados. A dissertação proporcionou uma visão de investigação, demonstrando o rigoroso procedimento adotado para alcançar resultados consistentes e passíveis de reprodução.

Em suma, a utilização do catálogo de ondas de rádio foi muito importante, pois nos possibilitou testar a isotropia do Universo em larga escala, através de dados de ondas de rádio, onde simulamos os mapas usando as bibliotecas Python, como matplotlib e Healpy, pela qual nos proporcionou simulações valiosas pois nos permitiu manipular e analisar os dados com eficiência.

# Referências

- 1 RYDEN, B. *Introduction to cosmology*. [S.l.]: Cambridge University Press, 2017. Citado 5 vezes nas páginas 5, 6, 10, 11 e 12.
- 2 AL, O. F. et. Citado 4 vezes nas páginas 5, 6, 18 e 33.
- 3 GORSKI, K. M. et al. Healpix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere. *The Astrophysical Journal*, IOP Publishing, v. 622, n. 2, p. 759, 2005. Citado 3 vezes nas páginas 7, 14 e 15.
- 4 MORISON, I. *Introduction to astronomy and cosmology*. [S.l.]: John Wiley & Sons, 2013. Citado 2 vezes nas páginas 10 e 11.
- 5 VIEGAS, S. M. *Descobrimos o Universo Vol. 56*. [S.l.]: Edusp, 2004. Citado na página 10.
- 6 DODELSON, S. *Modern cosmology*. [S.l.]: Elsevier, 2003. Citado na página 10.
- 7 HAWKING, S. *O universo numa casca de noz*. [S.l.]: Editora Intrínseca, 2016. Citado na página 10.
- 8 LIDDLE, A. *An introduction to modern cosmology*. [S.l.]: John Wiley & Sons, 2015. Citado 2 vezes nas páginas 11 e 12.
- 9 KARTTUNEN, H. et al. *Fundamental astronomy*. [S.l.]: Springer, 2007. Citado 3 vezes nas páginas 15, 16 e 17.
- 10 PERLEY, R. et al. The expanded very large array: A new telescope for new science. *The Astrophysical Journal Letters*, IOP Publishing, v. 739, n. 1, p. L1, 2011. Citado na página 17.
- 11 KASSIM, N. et al. Subarcminute resolution imaging of radio sources at 74 mhz with the very large array. *Astronomical Journal*, The University of Chicago Press for the American Astronomical Society, v. 106, p. 2218–2228, 1993. Citado na página 17.
- 12 CONDON, J. J. et al. The nrao vla sky survey. *The Astronomical Journal*, IOP Publishing, v. 115, n. 5, p. 1693, 1998. Citado 2 vezes nas páginas 18 e 19.