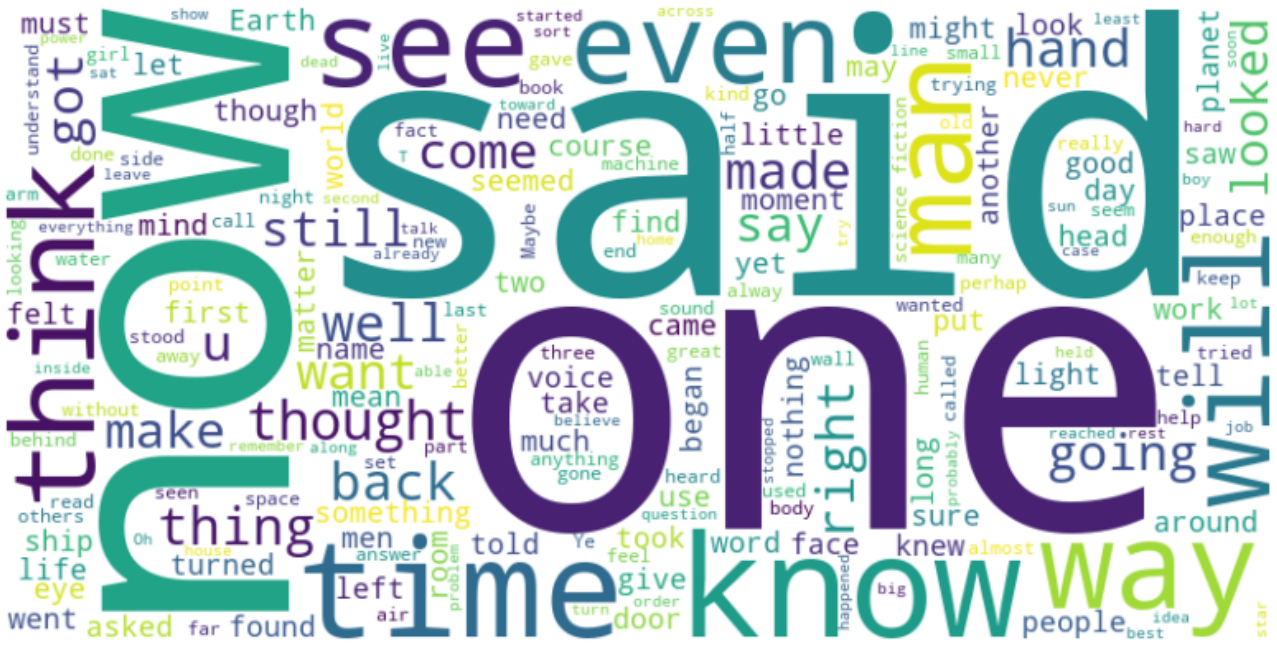
Creación de Diccionario

El text corpus se obtuvo de: https://www.kaggle.com/datasets/jannesklaas/scifi-stories-text-corpus

In [1]:
```
from collections import defaultdict
import re

def contar_palabras(texto):
    palabras = re.findall(r'\w+', texto.lower())
    contador = defaultdict(int)
    for palabra in palabras:
        contador[palabra] += 1
    return dict(contador)

# Cargar el archivo y contar las palabras
with open('internet_archive_scifi_v3.txt', 'r', encoding='utf-8') as archivo
    texto = archivo.read()
    histograma = contar_palabras(texto)

# Mostrar las 10 palabras más comunes
palabras_comunes = sorted(histograma.items(), key=lambda x: x[1], reverse=Tr
print(palabras_comunes)
```

```
[('the', 1525607), ('and', 674760), ('of', 652631), ('to', 640919), ('a', 63
2594), ('i', 423509), ('he', 418500), ('it', 380714), ('in', 365199), ('wa
s', 334168)]
```

In [4]:
```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

def crear_nube_de_palabras(texto):
    wordcloud = WordCloud(width=800, height=400, background_color='white').g
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    plt.show()

crear_nube_de_palabras(texto)
```



In [5]:
```python
def crear_diccionario(histograma):
    return set(histograma.keys())

diccionario = crear_diccionario(histograma)
```

In [12]:
```python
import textdistance

def corregir_palabra(palabra, diccionario):
    mejor_match = min(diccionario, key=lambda dicc_palabra: textdistance.lev
    return mejor_match

palabra_incorrecta = "tesgdb"
palabra_corregida = corregir_palabra(palabra_incorrecta, diccionario)
print(f"Corrección de '{palabra_incorrecta}': {palabra_corregida}")
```

```
Corrección de 'tesgdb': tesld
```