

ANÁLISE DE APONTAMENTO E JANELAS DE VISIBILIDADE PARA RASTREAMENTO DE SATÉLITES LEO

José Augusto Cevidanes Pedro

Instituto Federal São Paulo (IFSP), Cubatão, São Paulo, Brasil

Resumo: A proliferação de constelações de satélites de baixa órbita (LEO) impulsionou a demanda por estações terrestres ágeis e eficientes. Diferente de satélites geoestacionários, os satélites LEO possuem um movimento aparente rápido no céu, exigindo um rastreamento ativo e preciso que só é possível durante curtas janelas de visibilidade. Este trabalho apresenta o desenvolvimento de uma ferramenta computacional em *Python* para a predição de passagens de satélites LEO, servindo como base para a análise de requisitos operacionais de uma estação terrestre. Utilizando dados orbitais TLE (*Two-Line Element*) e o modelo de propagação SGP4, a ferramenta calcula os tempos de aquisição de sinal (AOS), máxima elevação e perda de sinal (LOS), bem como os vetores de apontamento (Azimute e Elevação). Como estudo de caso, foi realizada uma análise para a Estação Espacial Internacional (ISS) sobre a cidade de Santos-SP. A ferramenta demonstrou ser um recurso essencial para o planejamento de missões e para a especificação de subsistemas de controle e automação de antenas, permitindo a quantificação prévia dos requisitos de posicionamento do sistema.

Palavras-chave: Segmento Terrestre. Propagação Orbital. Controle de Antenas. *Software* de Engenharia. *Python*.

Abstract: The proliferation of Low Earth Orbit (LEO) satellite constellations has driven the demand for agile and efficient ground stations. Unlike geostationary satellites, LEO satellites have a fast apparent motion across the sky, requiring active and precise tracking that is only possible during short visibility windows. This work presents the development of a computational tool in Python for the prediction of LEO satellite passes, serving as a basis for the analysis of operational requirements for a ground station. Using Two-Line Element (TLE) orbital data and the SGP4 propagation model, the tool calculates the acquisition of signal (AOS), maximum elevation, and loss of signal (LOS) times, as well as the pointing vectors (azimuth and elevation). As a

case study, an analysis was performed for the International Space Station (ISS) over the city of Santos-SP. The tool proved to be an essential resource for mission planning and for the specification of antenna control and automation subsystems, allowing for the prior quantification of the system's positioning requirements.

Keywords: Ground Segment. Orbit Propagation. Antenna Control. Engineering Software. Python.

INTRODUÇÃO

A indústria espacial vive uma nova era, marcada pela ascensão de mega constelações de satélites de baixa órbita (*Low Earth Orbit* - LEO), que estão revolucionando setores como o de telecomunicações e o de sensoriamento remoto (Macambira, 2021). O sucesso técnico e econômico dessas constelações, no entanto, depende criticamente do segmento terrestre, responsável por manter a comunicação e o controle desses ativos espaciais (Prado; Kuga, 2001). No Brasil, estações de solo permanentes para comunicação com satélites de baixa órbita são praticamente inexistentes, o que indica uma subutilização de uma enorme quantidade de informações importantes disponibilizadas por estes satélites (Martins *et al.*, 2024).

O principal desafio de engenharia para sistemas LEO reside na sua dinâmica orbital. Ao contrário de um satélite de órbita geoestacionária (*Geostationary Earth Orbit* - GEO), que permanece fixo no céu, um satélite LEO atravessa o campo de visão de uma estação terrestre em minutos (Jesus *et al.*, 2017). Isso impõe a necessidade de sistemas de antenas com capacidade de rastreamento ativo, que devem apontar com precisão para o satélite durante toda a sua passagem visível (Souza, 2013). MEO, *Medium Earth Orbit*, é a órbita que fica entre a GEO e a LEO. O satélite MEO fica a 10.400 km da Terra, tem o período de revolução de 6 horas e uma vida média entre 7 a 10 anos.

A trajetória de um satélite pode ser calculada utilizando modelos de mecânica orbital, como o modelo geral de perturbações simplificadas (*simplified general perturbations 4* -SGP4), a partir de dados de elemento de duas linhas (*two-line element* – TLE) (Payne *et al.*, 2022). Ferramentas de software que implementam esses modelos são, portanto, o primeiro passo essencial para a operação de qualquer estação terrestre (Martins *et al.*, 2024).

A principal contribuição deste trabalho reside em apresentar uma metodologia de baixo custo, baseada em *software* de código aberto, para a quantificação de requisitos operacionais de estações terrestres, preenchendo uma lacuna entre a teoria da mecânica orbital e a prática do projeto de sistemas de controle para antenas. Através de um estudo de caso prático, focado no rastreamento da Estação Espacial Internacional (*international space station* -ISS) a partir de uma localização em Santos-SP, o artigo detalha a geração de um cronograma de passagens e discute como esses dados são fundamentais para o projeto de engenharia de uma estação de solo.

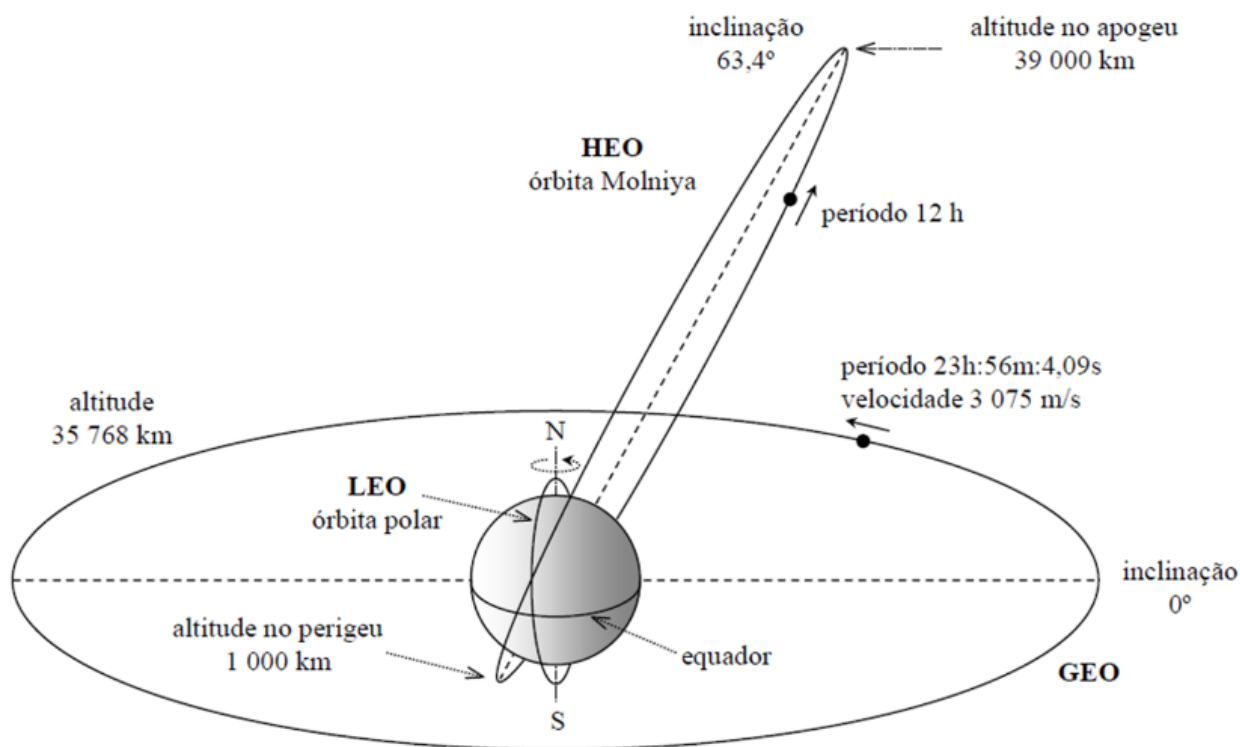
FUNDAMENTAÇÃO TEÓRICA

Para a correta compreensão da ferramenta desenvolvida, é necessário apresentar os conceitos fundamentais de mecânica orbital, os modelos matemáticos aplicados e os desafios de comunicação associados.

A. Sistemas de Coordenadas para Apontamento de Antenas

A posição de um satélite é geralmente descrita em um sistema de coordenadas geocêntrico, com origem no centro da Terra. No entanto, para uma estação terrestre, o que importa é a posição aparente do satélite no céu local. Essa posição é descrita pelo sistema de coordenadas topocêntrico, composto pelos ângulos de Azimute (Az) e Elevação (El) (Souza, 2013). Como ilustrado na Figura 1, a Elevação é o ângulo vertical do satélite em relação ao horizonte, enquanto o Azimute é o ângulo horizontal medido a partir do Norte geográfico. A tarefa de um *software* de rastreamento é, portanto, realizar a transformação de coordenadas do sistema geocêntrico para o topocêntrico para cada instante de tempo.

Figura 1 – Representação geométrica de uma passagem de satélite, com os ângulos de Azimute e Elevação.



Fonte: Adaptado de Academia de Redes (2025)

B. Elementos Orbitais de Duas Linhas (TLE)

Para que a posição de um satélite possa ser calculada, é preciso conhecer seus parâmetros orbitais. O formato padrão para a distribuição desses parâmetros é o TLE (*Two-Line Element set*). Um TLE é um conjunto de dados textuais que contém os elementos orbitais de um satélite em um formato padronizado, gerado e mantido pelo Comando de Defesa Aeroespacial da América do Norte (NORAD) (Flores; Saatkamp; Machado, 2017). A Figura 2 apresenta um exemplo da estrutura de dados contida em um arquivo TLE.

Figura 2 – Exemplo da estrutura de dados de um TLE (Two-Line Element).

The diagram illustrates the mapping of orbital parameters from a two-line element set (TLE) to specific fields in a type-4 data structure. Red circles and arrows link the parameters to their storage locations.

Top TLE Example:

```

1 60000U EXAMPLE 19001.06507538 +.00000000 00000 0 00000 0 0-0001
2 60000 4.6147 329.1013 0004698 305.1829 35.1588 1.002744140193

```

Bottom TLE Example:

```

1 60000U EXAMPLE 19001.06507538 +.00000000 29661-1 21450-1 4 0001
2 60000 4.6288 329.2287 0004409 309.5337 30.6658 1.0027453901937

```

Annotations:

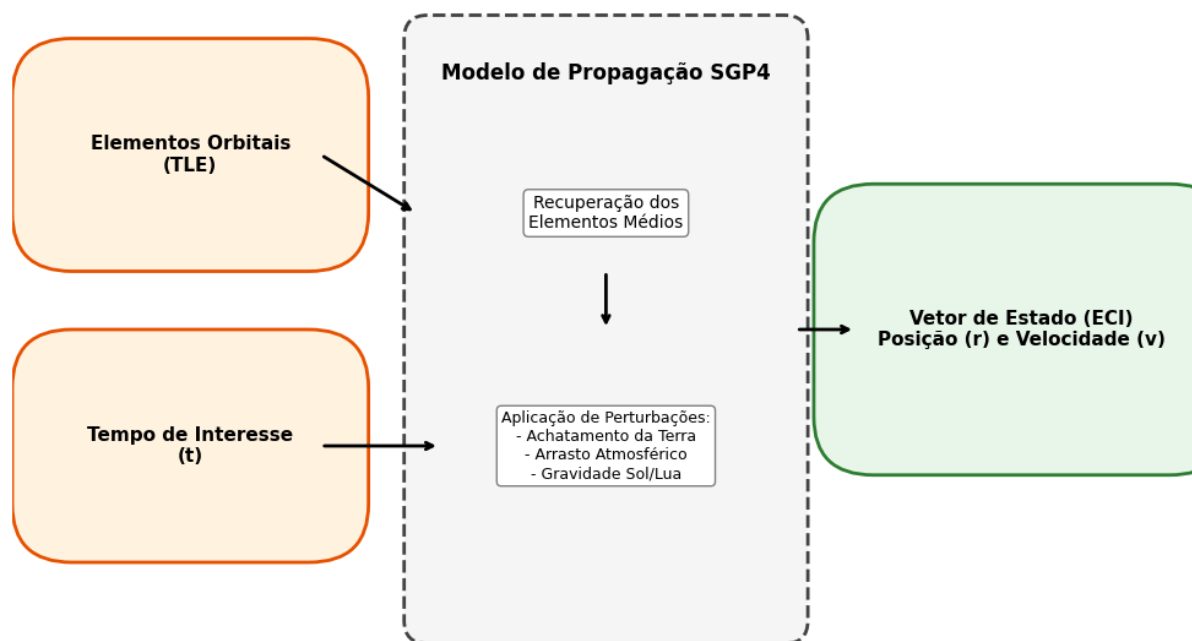
- Ephemeris Type:**
 - 0 – SGP/SGP4
 - 4 – SGP4-XP
- Solar Radiation Pressure:** When type-4, store SRP in $\ddot{n}/6$
- True Ballistic Coefficient:** When type-4, store B term in B* location
- Mean Motion:** When type-4, have Brouwer mean motion rather than Kozai

Fonte: Adaptado de KeepTrack (2025).

C. O Modelo SGP4 e as Perturbações Orbitais

A trajetória de um satélite LEO não é uma elipse perfeita, pois sofre perturbações constantes devido a fatores como o arrasto da atmosfera residual e as irregularidades no campo gravitacional da Terra (Prado; Kuga, 2001). O SGP4 é um modelo matemático de propagação orbital projetado especificamente para prever a posição e a velocidade de satélites próximos à Terra, levando em conta essas perturbações a partir de dados TLE (*Two-Line Element set*) (Payne et al., 2022). O modelo SGP4 foi desenvolvido por Ken Cranford em 1970. Este modelo foi obtido pela simplificação da teoria analítica mais extensa de Lane e Cranford que utiliza a solução de Brouwer para seu modelo gravitacional e uma função de densidade de potência para seu modelo atmosférico. A Figura 3 ilustra a estrutura simplificada de funcionamento do modelo SGP4, destacando suas entradas, o processamento das perturbações e os vetores de saída.

Figura 3 – Diagrama de blocos simplificado do modelo de propagação SGP4.



Fonte: O Autor (2025).

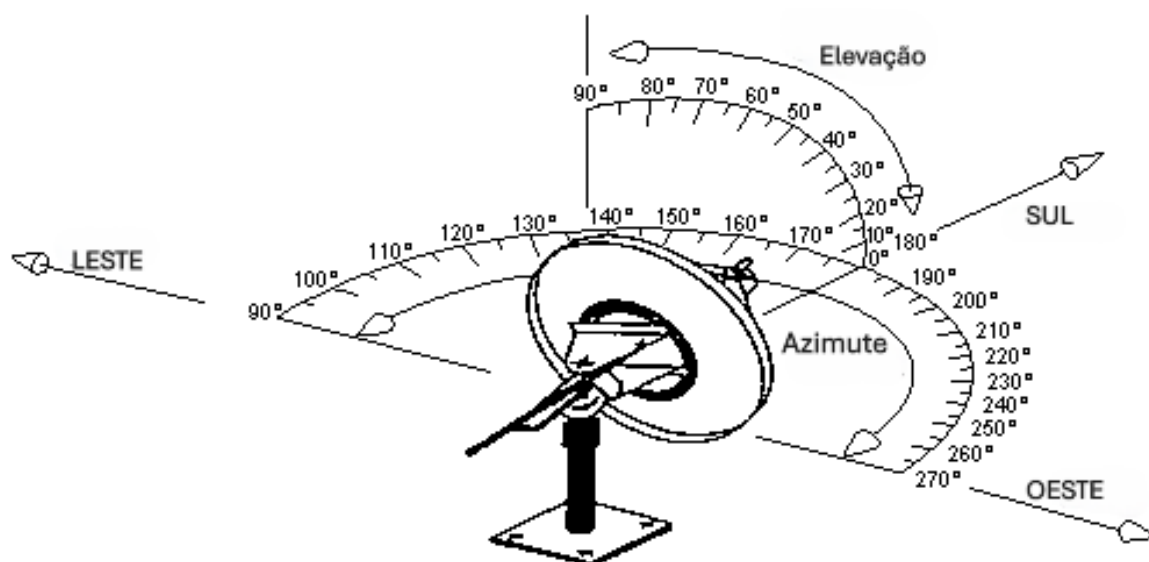
Este modelo fornece um balanço eficiente entre precisão e custo computacional, sendo o padrão utilizado pela grande maioria dos *softwares* de rastreamento de satélites (Flores; Saatkamp; Machado, 2017).

D. Desafios de Comunicação em Órbitas LEO

O rastreamento preciso é vital não apenas por questões geométricas, mas também por desafios impostos pela física das telecomunicações. Devido à alta velocidade relativa entre o satélite LEO e a estação terrestre (da ordem de 7,8 km/s), o sinal de rádio sofre uma variação significativa em sua frequência, fenômeno conhecido como Efeito *Doppler* (López-salamanca, 2020). Receptores de rádio precisam compensar ativamente essa variação para manter o sincronismo. Um apontamento impreciso da antena pode reduzir a relação sinal-ruído (*signal-to-noise ratio* - SNR) a ponto de o receptor perder o sinal, tornando a compensação de *Doppler* ineficaz. Além disso, a atenuação do sinal (perda de percurso) é máxima no início e no fim da passagem, quando o satélite está baixo no horizonte e a distância é maior. Portanto, um rastreamento preciso desde o primeiro instante de visibilidade é crucial para maximizar o tempo de comunicação útil. A Figura 4 ilustra um exemplo típico das

curvas de azimuth e elevação durante uma passagem, evidenciando a variação não linear e as altas taxas de variação (velocidade angular) que o sistema deve acompanhar.

Figura 4 – Exemplo de gráfico de trajetória para uma passagem de satélite, mostrando a variação do Azimute e da Elevação.



Fonte: Adaptado de Nova Eletrônica (2025).

E. O Rastreamento Ativo como um Problema de Controle

Do ponto de vista da Engenharia de Controle e Automação, o rastreamento de um satélite é um problema clássico de controle de servomecanismos. A sequência de valores de Azimute e Elevação calculada pela ferramenta de predição serve como a trajetória de referência, ou *setpoint*, que o sistema de controle da antena deve seguir. A capacidade do sistema em minimizar o erro entre a posição desejada (calculada pelo software) e a posição real da antena é o que garante a qualidade do enlace de comunicação. Portanto, a análise prévia da trajetória, especialmente o cálculo das velocidades e acelerações angulares máximas exigidas, é um passo fundamental na especificação dos atuadores (motores) e no projeto do controlador (Souza, 2013).

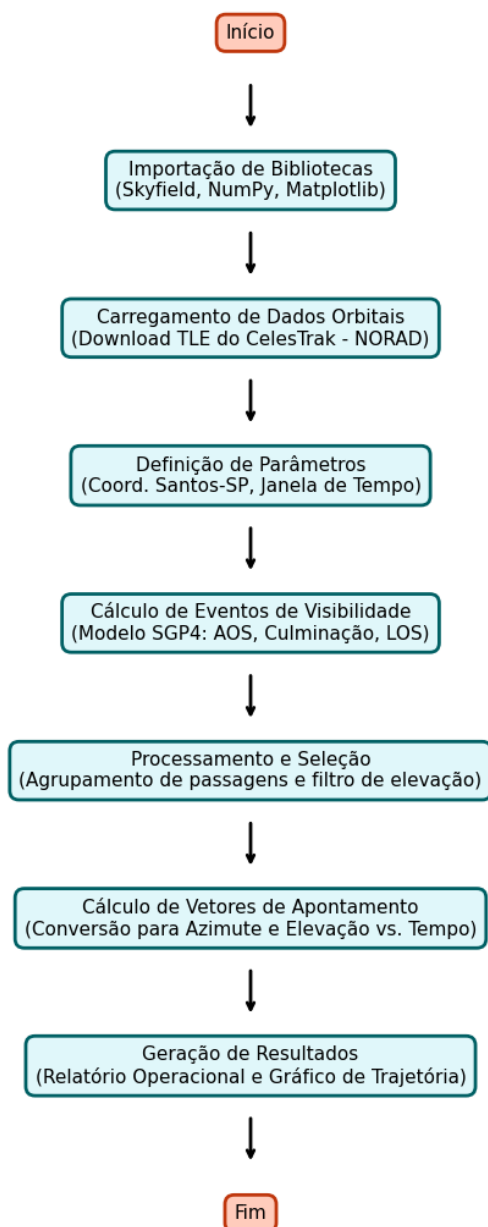
MATERIAL E MÉTODOS

O presente trabalho baseia-se no desenvolvimento de uma ferramenta computacional para predição de passagens de satélites e na sua aplicação em um estudo de caso.

A. Desenvolvimento da Ferramenta de Software

A ferramenta foi desenvolvida integralmente na linguagem de programação *Python*, utilizando a biblioteca *Skyfield*. A escolha por esta plataforma, em detrimento do uso isolado da biblioteca *SGP4* (ambas mantidas pelo mesmo desenvolvedor), deve-se à necessidade de conversão de sistemas de coordenadas. Enquanto a biblioteca *SGP4* fornece apenas os vetores de estado brutos no referencial inercial TEME (*True Equator, Mean Equinox*), a *Skyfield* integra esses cálculos com algoritmos astronômicos de alta precisão. Ela realiza automaticamente as complexas transformações vetoriais necessárias (considerando a rotação da Terra, nutação, precessão e as escalas de tempo UT1/UTC) para converter a posição do satélite em coordenadas topocêntricas (Azimute e Elevação) relativas ao observador em solo. Além disso, por ser de código aberto, alinha-se com a proposta de um projeto de baixo custo (Payne et al., 2022). Para facilitar a compreensão da lógica implementada, a Figura 5 apresenta o diagrama de blocos com o fluxo de execução do algoritmo desenvolvido.

Figura 5 – Fluxograma do algoritmo de rastreamento desenvolvido.



Fonte: Elaborado pelo autor (2025).

O processo detalhado na Figura 5 segue os seguintes passos principais:

1. **Carregamento de Dados Orbitais:** A ferramenta inicia conectando-se a um repositório público (*CelesTrak*) para baixar o arquivo TLE mais recente para o satélite de interesse.
2. **Definição do Observador:** As coordenadas geográficas da estação terrestre (Santos-SP) são definidas no padrão WGS84.

3. **Janela de Tempo:** O período de análise é definido para as próximas 24 horas a partir do momento da execução do *script*.
4. **Cálculo de Eventos:** É determinada a sequência de instantes em que o satélite cruza um limiar de Elevação mínimo, definido em 10 graus acima do horizonte, calculando-se o início (AOS), o ponto de máxima elevação e o fim (LOS) de cada passagem.
5. **Geração do Relatório:** Para cada evento, o *software* determina o vetor de apontamento (Azimute e Elevação) e formata os dados em um relatório operacional.

B. Estudo de Caso

Para validar a ferramenta, foi conduzido um estudo de caso focado no rastreamento de dois satélites com perfis orbitais distintos. A escolha se deu por ser um objeto de grande interesse com uma órbita de baixa inclinação (aproximadamente 51.6°), representativa de muitas missões tripuladas e de observação. Em contraponto, foi selecionado o satélite NOAA 20 (JPSS-1), que possui uma órbita polar (inclinação próxima a 98°), típica de satélites meteorológicos e de sensoriamento remoto. Esta escolha estratégica permite demonstrar a flexibilidade da ferramenta em analisar diferentes cenários de missão e validar seu funcionamento para uma gama mais ampla de satélites LEO.

RESULTADOS E DISCUSSÃO

A execução da ferramenta de *software* descrita na seção anterior, configurada para a estação terrestre em Santos-SP, produziu o relatório de passagens visíveis da Estação Espacial Internacional (ISS) para um período específico de 24 horas. Para fins deste estudo de caso, foi simulada a operação para o dia 15 de outubro de 2025. Os dados operacionais resultantes, que variam diariamente devido à precessão da órbita, estão consolidados e apresentados na Tabela 1.

Tabela 1 – Previsão de passagens da ISS para a estação terrestre em Santos-SP para o dia 15/10/2025 (Elevação mínima > 10°).

Passage m	Evento	Horário (Local)	Azimute	Elevação	Duração
1	Início (AOS)	02:28:34	179.2° (S)	-	4 min 29 seg
1	Máxima Elevação	02:30:49	137.6° (SE)	16.4°	4 min 29 seg
1	Fim (LOS)	02:33:03	95.9° (E)	-	4 min 29 seg
2	Início (AOS)	04:04:40	247.9° (WSW)	-	5 min 43 seg
2	Máxima Elevação	04:07:32	306.3° (NW)	25.8°	5 min 43 seg
2	Fim (LOS)	04:10:23	4.7° (N)	-	5 min 43 seg
3	Início (AOS)	17:27:12	346.9° (NNW)	-	6 min 8 seg
3	Máxima Elevação	17:30:15	53.0° (NE)	33.2°	6 min 8 seg
3	Fim (LOS)	17:33:20	118.9° (ESE)	-	6 min 8 seg
4	Início (AOS)	19:05:17	253.1° (WSW)	-	3 min 33 seg
4	Máxima Elevação	19:07:03	221.2° (SW)	13.5°	3 min 33 seg
4	Fim (LOS)	19:08:50	189.4° (S)	-	3 min 33 seg

Fonte: Elaborado pelo autor (2025).

Nota: AOS (*Acquisition of Signal*) indica o início da visibilidade; LOS (*Loss of Signal*) indica o fim da visibilidade. O ponto de Máxima Elevação corresponde ao TCA (*Time of Closest Approach*). A elevação para AOS e LOS refere-se ao limiar mínimo de 10°.

A análise dos dados da Tabela 1 permite extrair discussões relevantes para a engenharia. A principal delas está relacionada aos requisitos impostos ao sistema de controle e automação da antena. Passagens mais altas, como a Passagem 3 (máx. 33.2°), exigem que o sistema de controle execute maiores variações angulares em um curto período de tempo. A análise da trajetória desta passagem revela uma variação de Azimute de aproximadamente 66 graus em cerca de 6 minutos, exigindo uma velocidade angular média de 0.18 graus/segundo. Esses dados são cruciais, pois impactam diretamente na escolha dos atuadores do sistema de controle (Souza, 2013).

Somando as durações das quatro passagens, o tempo total de contato com a ISS em 24 horas é de aproximadamente 20 minutos. Esta disponibilidade limitada é uma característica intrínseca de satélites LEO individuais e justifica o uso de grandes constelações para se obter cobertura contínua (Macambira, 2021). Para demonstrar a versatilidade da ferramenta, uma análise similar para o satélite NOAA 20 revelou passagens de Elevação consistentemente mais altas, validando a ferramenta como um instrumento flexível para o planejamento de missões. O código fonte utilizado está no Apêndice e pode estar publicamente disponível na referência (Pedro, 2025).

CONCLUSÕES

Este trabalho apresentou o desenvolvimento e a aplicação de uma ferramenta computacional em Python para a predição de passagens de satélites LEO, abordando um desafio central para a operação de estações terrestres. O objetivo de criar uma solução de baixo custo para gerar dados operacionais de apontamento foi alcançado com sucesso.

A metodologia empregada demonstrou ser robusta e eficaz. A discussão dos resultados evidenciou a importância da ferramenta como um instrumento de análise para a engenharia de sistemas, permitindo quantificar requisitos críticos para o sistema de controle da antena e analisar a disponibilidade do enlace. Conclui-se que o software proposto constitui uma base sólida para o planejamento, projeto e operação de estações terrestres de baixo custo, contribuindo para viabilizar a exploração de dados de satélites LEO no Brasil (Martins *et al.*, 2024).

Como trabalhos futuros, sugerem-se as seguintes expansões para o projeto:

- a) A integração da ferramenta com um sistema de controle de hardware em malha fechada, permitindo o apontamento automático de um protótipo de antena;
- b) A expansão do software para analisar constelações inteiras, calculando a disponibilidade de comunicação total e os eventos de handover entre satélites;
- c) A implementação de uma interface gráfica para o usuário (GUI), tornando a ferramenta mais acessível para fins educacionais e operacionais.

REFERÊNCIAS

ACADEMIA DE REDES. Rede Satelital. Disponível em: <https://academiaderedes.com/conteudos/wwan/rede-satelital/>. Acesso em: 11 out. 2025.

FLORES, R. D.; SAATKAMP, E. D.; MACHADO, R. Protótipo de uma estação receptora de imagens meteorológicas utilizando GNU Radio e RTL-SDR. *In: SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES E PROCESSAMENTO DE SINAIS*, 35., 2017, São Pedro. Anais do XXXV Simpósio Brasileiro de Telecomunicações e Processamento de Sinais.

JESUS, A. S. et al. Sistema Automático de Apontamento de Antena para Receptores Satelitais nas Bandas C e Ku. *In: SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES E PROCESSAMENTO DE SINAIS*, 35., 2017, São Pedro. Anais do XXXV Simpósio Brasileiro de Telecomunicações e Processamento de Sinais.

KEEPTRACK.SPACE. SGP4-XP Deep Dive. Disponível em: <https://keeptrack.space/deep-dive/sgp4-xp>. Acesso em: 19 nov. 2025.

LÓPEZ-SALAMANCA, J. J. Modelo para caracterização de canal através de Cadeias de Markov de um sistema de comunicação entre CubeSat e estação terrestre. 2020. Tese (Doutorado em Engenharia Elétrica) – Universidade Federal de Santa Catarina, Florianópolis, 2020.

MACAMBIRA, R. N. M. Otimização do consumo de energia em uma rede de satélites de baixa órbita em cenários de alta vazão. 2021. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal do Amazonas, Manaus, 2021.

MARTINS, A. L. et al. Uma proposta de estações de solo para satélites artificiais de baixa órbita. *In: CONGRESSO DE INOVAÇÃO, CIÊNCIA E TECNOLOGIA DO IFSP*, 15., 2024. Anais do 15º Congresso de Inovação, Ciência e Tecnologia do IFSP.

NOVA ELETRÔNICA. Programa Online de Apontamento de Antena Parabólica. Disponível em: <https://blog.novaeletronica.com.br/programa-online-de-apontamento-de-antena-parabolica/>. Acesso em: 11 out. 2025.

PAYNE, T. et al. Improvements to the SGP4 propagator (SGP4-XP). *In: ADVANCED MAUI OPTICAL AND SPACE SURVEILLANCE TECHNOLOGIES CONFERENCE*, 2022. Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference.

PEDRO, José Augusto Cevidanes. Artigo-qualif-rastreador-satélite, Github, 2025. Disponível em: <https://github.com/joseaugustocpedro/artigo-qualif-rastreador-satelite>. Acesso em 12 nov. 2025.

PRADO, A. F. B. A.; KUGA, H. K. (Org.). Fundamentos de tecnologia espacial. São José dos Campos: INPE, 2001.

RHODES, B. SGP4: Track earth satellites using the SGP4 standard. PyPI, 2024. Disponível em: <https://pypi.org/project/sgp4/>. Acesso em: 21. nov. 2025.

RHODES, B. Skyfield: Elegant Astronomy for Python. PyPI, 2024. Disponível em: <https://pypi.org/project/skyfield/>. Acesso em: 21. nov. 2025.

SOUZA, A. L. G. Sistema de apontamento de antena para estação de comunicação com satélites. 2013. Dissertação (Mestrado em Engenharia Elétrica) – Universidade de Brasília, Brasília, 2013.

APÊNDICE A – CÓDIGO-FONTE DA FERRAMENTA DE RASTREAMENTO EM PYTHON

O código a seguir apresenta a implementação completa da ferramenta desenvolvida, responsável pela conexão com o repositório CelesTrak, cálculo dos eventos orbitais e geração do relatório de passagens.

```
# -*- coding: utf-8 -*-
from datetime import timedelta
import pytz
import os

# Tenta importar as bibliotecas e dá uma mensagem clara se não estiverem
instaladas
try:
    from skyfield.api import load, wgs84
except ImportError:
    print("ERRO: Biblioteca 'skyfield' não encontrada.")
    print("Por favor, instale-a executando o comando no seu terminal: pip
install skyfield")
    exit()

def format_azimuth(degrees):
    """Converte um ângulo em graus para uma direção cardeal (N, NNE, NE,
etc.)."""
    dirs = ["N", "NNE", "NE", "ENE", "E", "ESE", "SE", "SSE",
            "S", "SSW", "SW", "WSW", "W", "WNW", "NW", "NNW"]
    index = round(degrees / (360. / len(dirs)))
    return dirs[index % len(dirs)]

def run_satellite_tracker():
    """
    Função principal que calcula e imprime as passagens de um satélite
    sobre uma localização específica.
    """
    try:
        # --- 1. CONFIGURAÇÃO ---
        # Usa a forma padrão da biblioteca, salvando dados na pasta do
usuário.
        ts = load.timescale()

        # URL do arquivo com os TLEs das estações espaciais
        url =
'https://celestrak.org/NORAD/elements/gp.php?GROUP=stations&FORMAT=tle'

        print("-> Baixando os dados mais recentes dos satélites de
celestrak.org...")
        sats = load.tle_file(url, reload=True)
        print(f"    {len(sats)} satélites carregados com sucesso.")

        # Procura pela Estação Espacial Internacional (ISS)
        satellite_name = 'ISS (ZARYA)'
        satellite = next((s for s in sats if s.name == satellite_name), None)

        if satellite is None:
```

```

print(f"ERRO: Satélite '{satellite_name}' não encontrado na
lista.")
return

# Define a localização da estação terrestre (Santos, SP)
ground_station = wgs84.latlon(-23.9608, -46.3322, elevation_m=10)
local_tz = pytz.timezone('America/Sao_Paulo')

# --- 2. DEFINIÇÃO DO PERÍODO DE ANÁLISE ---
t0 = ts.now()
t1 = t0 + timedelta(days=1)

# --- 3. CÁLCULO DOS EVENTOS ---
print(f"-> Calculando as passagens da '{satellite.name}' sobre
Santos...")
times, events = satellite.find_events(ground_station, t0, t1,
altitude_degrees=10.0)

if not events.any():
    print("\nNenhuma passagem visível da ISS nas próximas 24 horas.")
    return

# --- 4. FORMATAÇÃO E EXIBIÇÃO DO RELATÓRIO ---

print("\n=====
=====")
print(f"[ PREVISÃO DE PASSAGENS PARA {satellite.name} SOBRE SANTOS,
SP ]")
print(f"Localização: Lat {ground_station.latitude.degrees:.2f}°, Lon
{ground_station.longitude.degrees:.2f}°")
print(f"Período: Próximas 24 horas a partir de
{t0.astimezone(local_tz).strftime('%Y-%m-%d %H:%M:%S')}")

print("=====
==")

pass_events = []
# Agrupa os eventos em passagens individuais
for t, event in zip(times, events):
    if event == 0: # Evento de "nascer" do satélite inicia uma nova
passagem
        pass_events.append([])
        if pass_events: # Garante que a lista não está vazia
            pass_events[-1].append((t, event))

pass_num = 1
for pass_group in pass_events:
    event_types = [e[1] for e in pass_group]
    # Verifica se a passagem é completa (contém os 3 eventos: nascer,
culminar, se pôr)
    if 0 in event_types and 1 in event_types and 2 in event_types:
        t_rise = pass_group[0][0]
        t_culminate = pass_group[1][0]
        t_set = pass_group[2][0]

        alt_r, az_r, _ = (satellite -
ground_station).at(t_rise).altaz()
        alt_c, az_c, _ = (satellite -
ground_station).at(t_culminate).altaz()

```



```
        alt_s, az_s, _ = (satellite -
ground_station).at(t_set).altaz()
        duration = (t_set - t_rise) * 24 * 60

        print(f"\n-> Passagem {pass_num}:")
        print(f"                Início da Visibilidade (AOS):
{t_rise.astimezone(local_tz).strftime('%H:%M:%S')} | Azimute:
{az_r.degrees:.1f}° ({format_azimuth(az_r.degrees)})")
        print(f"                Ponto de Máxima Elevação:
{t_culminate.astimezone(local_tz).strftime('%H:%M:%S')} | Azimute:
{az_c.degrees:.1f}° ({format_azimuth(az_c.degrees)}) | Elevação:
{alt_c.degrees:.1f}°")
        print(f"                Fim da Visibilidade (LOS):
{t_set.astimezone(local_tz).strftime('%H:%M:%S')} | Azimute:
{az_s.degrees:.1f}° ({format_azimuth(az_s.degrees)})")
        print(f"        Duração Total da Passagem: {int(duration)} min
{int((duration % 1) * 60)} seg")
        pass_num += 1

    except Exception as e:
        print(f"\n--- OCORREU UM ERRO INESPERADO ---")
        print(f"Mensagem: {e}")

if __name__ == "__main__":
    run_satellite_tracker()
```