Flame Language - Technical Whitepaper

1. Introduction

Flame is a domain-specific programming language developed by Daniel Ricardo Maranhao Santana, under the institutional support of Anternative 3 and Wareness. Its core purpose is to simplify and accelerate data-driven automation, especially within environmental and geospatial analysis systems.

Flame was inspired by the need to bridge simplicity and expressiveness for real-time decision-making in civil protection contexts.

2. Design Philosophy

Flame's syntax draws from Python, but with a minimalist design tailored for scripting and clarity. Key goals include:

- Fast onboarding for field professionals
- Seamless integration with geospatial and meteorological systems
- Expressive syntax with strong focus on automation and control
- Compatibility with Streamlit, Python backends and AI models

3. Syntax Overview

Basic operations are intuitive:

- Variable assignment: let x = 5
- Output: print("x is", x)
- Loops: for i in 1..10:
   print(i)
- Conditions: if x > 5:
   print("high")
- Functions: def sum(a, b):
   return a + b

4. Use Cases

Flame was conceived for:

- Automating meteorological data processing
- Fire behavior simulation and prediction
- Generating geospatial reports by region/concelho
- Orchestrating scripts on NAS servers or cloud deployments
- Supporting environmental monitoring dashboards

5. Integration

Flame scripts can be executed directly, embedded in Streamlit dashboards, or integrated in AI pipelines. They support reading/writing structured data, calling Python functions, and invoking predictive models trained in scikit-learn or PyTorch.

6. License and Distribution

Flame is released under the MIT License. Open to academic, operational, and community contributions via GitHub. Official repository: https://github.com/dsantananet/flame-lang

7. Acknowledgements

Developed by Daniel Ricardo Maranhao Santana.

Institutional support: Anternative 3

Endorsed by: Wareness