



Business Requirements Document

2025

Documento de Requisitos de Negócio (BRD) - Python

1. Cabeçalho e Identificação

- Título do Documento: BRD – Fluxo Zero
- Nome da empresa: Fluxo Zero
- Autores:
Azor Biagioni Tartuce, Programador de BackEnd e FrontEnd. RM: 563995;
Daniel Oliveira de Souza, Programador de BackEnd e FrontEnd. RM: 566284;
Lucas de Almeida Pires, Analista de Usabilidade e Gestor de Audiovisual. RM: 562757;
- Versão atual do documento: 1.0
- Data da criação do documento: 27/05/2025
- Aprovado por: professor Aurélio, supervisor acadêmico do projeto.

2. Requisitos Detalhados do Programa em Python

2.1 Requisitos Funcionais (RF)

- Python:
 - RF01: O programa deve requisitar dados como histórico do local, chuva atual ou prevista, solo e sua permeabilidade, relevo e nível de infraestrutura urbana;
 - RF02: O aplicativo deve analisar os dados informados e simular insights a partir deles;
 - RF03: O aplicativo deve mostrar os resultados simulados;
 - RF04: O aplicativo deve mostrar insights sobre possíveis impactos da catástrofe;
 - RF05: O aplicativo deve mostrar insights sobre medidas para evitar os problemas;
 - RF06: O aplicativo deve ter um histórico de usos (salvo em JSON, TXT ou CSV), em que é possível visualizar dados antigos.
- Python:
 - RN01: O aplicativo deve operar continuamente durante sua execução, permitindo reinicializações ilimitadas sem perda de desempenho;

- RN02: O aplicativo deve ser capaz de identificar e notificar o usuário em caso de inserção de dados incorretos ou inconsistentes com a realidade;
- RN03: O aplicativo deve informar caso o usuário escolha uma opção inválida;
- Python:
 - RNF01: O aplicativo deve estar livre de falhas críticas (bugs) que comprometam sua funcionalidade principal;
 - RNF02: O aplicativo deve ser otimizado para uso eficiente de memória, garantindo desempenho fluido mesmo em dispositivos com recursos limitados;
 - RNF03: O aplicativo deve ter como requisito mínimo 512MB de memória RAM e 50MB de espaço em disco;
 - RNF04: O aplicativo deve ser compatível e funcional em sistemas operacionais desktop amplamente utilizados, como Windows, Linux e MacOS.
- Python:
 - O programa deve solicitar respostas ao usuário com perguntas objetivas sobre a região;
 - As respostas devem gerar uma análise com base em lógica condicional clara e coerente;
 - O sistema deve alertar o usuário em caso de risco de enchente, com base nas respostas fornecidas;
 - O resultado deve apresentar insights e orientações com linguagem compreensível;
 - O programa deve permitir múltiplas execuções sem travamentos ou perda de desempenho;
 - Em caso de entrada inválida (ex: texto onde deveria ser número), o sistema deve exibir mensagem de erro clara e seguir operando;
 - O script deve utilizar menos de 512MB de RAM durante execução padrão;
 - O histórico de respostas deve ser salvo em arquivo externo para futuras análises.