**Documento de Requisitos**

**Clip-cli**

**Matheus Brito**  
Engenheiro de Software

**Sumario**

**1. Introdução**

**1.1.** Proposito do Documento

**1.2.** Escopo do Produto

**1.3.** Definição e Abreviações

**1.4.** Visão Geral do Documento

**2. Descrição Geral**

**2.1.** Perspectiva do Produto

**2.2.** Funções do Produto

**2.3.** Restrições Gerais

**3. Requisitos**

**3.1.** Requisitos Funcionais

**3.2.** Requisitos não Funcionais

**3.3.** Outros Requisitos

**4. Atributos**

**4.1.** Disponibilidade

**4.2.** Segurança

**4.3.** Manutenção

**1. Introdução**

**1.1. Propósito do Documento**

Este documento contém a especificação de requisitos para a CLI Clip, que gerenciara vários processos de desenvolvimento em um único terminal.

**1.2. Escopo do Produto**

O sistema tem como objetivo auxiliar no gerenciamento dos serviços necessários para o ambiente de desenvolvimento, como: react, processamento de sass, inicialização de docker, e vários outros..

**1.3 Definições e Abreviações**

As definições utilizadas neste documento serão abordadas posteriormente no glossário. Abreviações: • *RF: requisito funcional;* • *RNF: requisito não funcional*

**1.4 Visão Geral do documento**

Este documento apresenta uma descrição geral da cli, e logo em seguida descreve suas funcionalidades especificando as entradas e saídas para todos os requisitos funcionais. Faz também uma descrição sucinta dos requisitos não funcionais contidos neste sistema.

**2. Descrição Geral**

A cli CLIP gerencia os processos dentro do ambiente de desenvolvimento, onde o desenvolvedor terão permissão para inserir, modificar, excluir, executar ambientes e seus respectivos comandos.

**2.1. Perspectiva do Produto**

A cli opera na máquina do desenvolvedor que gerencia um arquivo JSON e controla a saída de cada comando do ambiente em um único arquivo de log.

**2.2. Funções do Produto**

Gerenciamento de ambientes de desenvolvimento: inserir, modificar, excluir, executar ambientes de desenvolvimento.

**2.3. Restrições Gerais**

A cli depende da instalação do Nodejs e seu gerenciador de pacotes o NPM.

**3. Requisitos**

**3.1. Requisitos Funcionais**

*RF. 1: Cadastro de ambiente.*

Descrição: Caso o ambiente informado já exista apenas o comando será adicionado ao ambiente.

Entrada: Nome do ambiente, diretório, comando.

Processo: Caso o ambiente informado já exista apenas o comando será adicionado ao ambiente e armazenado no arquivo de configuração JSON.

Saída: Mensagem de confirmação bem sucedido do cadastro caso tenha sido efetuado com sucesso, senão, mensagem de erro.

*RF. 2: Modificação de comando de ambiente*

Descrição: O usuário entra com o ambiente e comando que deseja modificar e o modifica. Entrada: nome do ambiente e posição do comando

Processo: Atualização do comando no arquivo de configuração JSON.

Saída: Mensagem de confirmação bem sucedido da modificação do cadastro caso tenha sido efetuado com sucesso, senão, mensagem de erro.

*RF 3: Exclusão do ambiente ou comando*

Descrição: O usuário poderá excluir o ambiente ou comando.

Entrada: Nome do ambiente ou nome do ambiente mais posição do comando para excluir comando especifico do ambiente.

Processo: A cli verifica se o ambiente ou comando existe, se sim é excluído.

Saída: Mensagem de confirmação bem sucedido da exclusão do cadastro caso tenha sido efetuado com sucesso, senão, mensagem de erro.

*RF. 4: Executar ambiente*

Descrição: Os usuários cadastrados podem inserir documentos com suas descrições.

Entrada: Autor(es), título, palavras-chaves, resumo, local de aplicação, upload dos documentos. Processo: O sistema insere todos esses dados no no banco de dados.

*RF.5: Visualizar ambientes*

Descrição: O usuário pode visualizar todos os ambientes e seus respectivos comandos cadastrados.

Processo: Busca o arquivo de configurações JSON todos os ambientes e seus respectivos comandos e exibe para o usuário.

**3.2- Requisitos Não Funcionais**

**3.2.1-Requisitos Organizacionais**

*RNF. 1: Software.*

O SGBD utilizado será o PostgreSQL 8.2.4. Este SGBD apesar de ser gratuito é muito confiável.

*RFN. 2: Linguagem de Programação*

A cli será feita utilizando NodeJs.

*RFN. 3: Instalação*

A cli será distribuída e instala com o NPM (gerenciador de pacotes do Nodejs).

**3.3-Outros Requisitos**

O sistema funcionará somente em sistemas operacionais Windows.

**4-Atributos**

**4.1-Disponibilidade**

A cli deve estar sempre disponível, caso ocorra alguma interrupção ele deve ser restaurado o mais rápido possível.

**4.2-Segurança**

A maior preocupação é com atualizações de segurança para futura versões do NodeJs e seus módulos nativos.

**4.3-Manutenção**

A manutenção será feita por Matheus B. responsável pelo desenvolvimemento desse projeto.