

Documento de Requisitos

Responsável

Alysson Gomes de Sousa

**Quixadá
2017**

Histórico de Revisão

Data	Versão	Descrição	Autor
30/05/2017	1.0	Versão inicial do documento.	Alysson Gomes de Sousa
04/06/2017	2.0	Inclusão dos requisitos funcionais e não funcionais	Alysson Gomes de Sousa

1. Introdução

Este documento tem o objetivo de especificar os requisitos do sistema de apoio a desenvolvedores e usuários da plataforma *HPC-Shelf*, denominado Interactive *SHELF* (*IShelf*).

2. Descrição do Problema

O problema consiste na inexistência de um sistema de apoio à desenvolvedores e usuários de aplicações de computação de alto desempenho na plataforma *HPC-Shelf*.

3. Descrição Geral do Sistema

O sistema *IShelf*, é uma ambiente *web* que funciona como camada de *software* que atua sobre a plataforma *HPC-Shelf*, e deverá prover funcionalidades para criação e gerenciamento de componentes, aplicações e plataformas de execução de forma interativa e intuitiva.

4. Finalidade

O sistema *IShelf* permitirá que desenvolvedores e especialistas da computação de alto desempenho tenham a capacidade de criar aplicações com alta demanda computacional em um plataforma em nuvem de forma amigável e interativa.

5. Convenções e Termos

5.1. Identificação dos requisitos

Por convenção, os requisitos serão referenciados por siglas, sendo [RFXX] para requisitos funcionais [RNFXX] para requisitos não-funcionais e [RDXX] para requisitos de domínio, sendo 'XX' a representação de uma sequência numérica iniciando de '01' para cada categoria de requisitos.

5.2. Prioridade dos requisitos

Para estabelecer as prioridades dos requisitos em relação ao sistema como um todo, foram adotadas as seguintes denominações: **Baixa**, **Média** e **Alta**.

Baixa: requisitos com essa prioridade não afetam a funcionalidade básica do sistema, ou seja, o sistema pode funcionar de forma satisfatória sem eles. Esses requisitos podem ser implementados em futuras versões do sistema, caso o tempo de desenvolvimento seja curto.

Média: O sistema funciona sem esse requisito, mas de forma não satisfatória. Esses requisitos devem ser implementados na versão inicial do sistema, mas se não forem, o sistema pode ser implantado sem eles.

Alta: São os requisitos sem os quais o sistema não entra em funcionamento. O sistema não pode ser implantado sem esses requisitos.

6. Requisitos

Nesta seção os requisitos são apresentados, eles estão divididos em três subseções, Requisitos funcionais, Requisitos não-funcionais e Requisitos de domínio.

6.1. Requisitos funcionais

Esses requisitos são refletidos como funcionalidades no sistema, eles descrevem exatamente como o sistema deve funcionar operar. Os requisitos do sistema de agendamento são descritos a seguir:

Identificador	Prioridade	Descrição
RF01	Alta	O sistema deve conter um controle de acesso para distinguir cada tipo de usuário.
RF02	Alta	O sistema deve permitir que um usuário do tipo desenvolvedor, possa criar e editar um componente abstrato.
RF03	Alta	O sistema deve listar os componentes disponíveis.
RF04	Media	Para cada componente selecionado, o sistema deve mostrar as informações do componente escolhido.
RF05	Media	O sistema deve permitir que o desenvolvedor inclua um ou mais funcionalidades no

		componente, representado por uma unidade abstrata.
RF06	Media	O sistema deve permitir que outros componentes possam ser associados ao componente criado.
RF07	Media	O sistema deve permitir que o desenvolvedor inclua características no componente criado, representadas por parâmetros de contexto, de qualidade, de custo e de ranking.
RF08	Baixa	O sistema deve permitir que o desenvolvedor faça o <i>upload</i> de arquivos de código-fonte para uma determinada unidade.
RF09	Alta	O sistema deve permitir que instâncias (contratos contextuais) dos componentes possam ser criadas. Além de permitir que esta instância seja salva em um arquivo para o desenvolvedor.
RF10	Alta	O sistema deve permitir que as instâncias dos componentes possam ser executadas. Além disso, deve permitir que o resultado do processamento da instância seja salvo em um arquivo.
RF11	Alta	O sistema deve permitir que componentes de tipos específicos possam ser criados (Ex.: componente da espécie plataforma).
RF12	Media	O sistema deve permitir que os montadores possam criar a arquitetura de aplicações baseadas em componentes.
RF13	Media	O sistema deve permitir que o fluxo dos <i>workflows</i> possa ser definido a partir de sua arquitetura.

6.2. Requisitos não-funcionais

Esses requisitos são refletidos como restrições e qualidade do sistema, e são descritos a seguir:

Identificador	Descrição
RNF01	O sistema deverá ser desenvolvido na linguagem de programação Java e JavaScript, usando paradigma orientado a objetos juntamente com o padrão MVC.
RNF02	O sistema deverá ser uma aplicação web.
RNF03	A comunicação com a plataforma HPC- <i>Shelf</i> deve ser feita através do <i>web service</i> através da linguagem XML.

6.3. Requisitos de domínio

Esses requisitos são refletidos como características do domínio do sistema, e são descritos a seguir:

Identificador	Descrição
RD01	Cada componente deve possuir um super componente (componente pai)

7. Casos de Uso do Projeto

7.1. Atores

Identificador	Ator	Descrição
A01	Desenvolvedor	Desenvolvedor de componentes abstratos.
A02	Montador	Montador de aplicações..
A03	Mantenedor	Mantenedor das plataformas físicas de execução.

7.2. Casos de Uso

Identificador	Caso de Uso	Descrição
UC01	Login na plataforma	Os usuários deverão realizar login para ter acesso ao sistema.

UC02	Manter componentes	O usuário poderá cadastrar e editar componentes.
UC03	Criar unidades abstratas	O usuário poderá adicionar unidades abstratas no componentes que está criando.
UC04	Criar parâmetros de contexto	O usuário poderá adicionar parâmetros de contexto no componentes que está criando.
UC05	Adicionar componentes aninhados	O usuário poderá adicionar outros componentes existentes no componentes que está criando.
UC06	Selecionar componentes	O usuário poderá selecionar um componente já existente para editá-lo ou adicioná-lo ao componente que está criando.
UC07	Criar contrato contextual	O usuário poderá criar uma instância de um determinado componente previamente selecionado pelo mesmo.
UC08	Realizar <i>download</i> do contrato contextual	Após a criação da instância, o usuário poderá salvar em um arquivo.
UC09	Realizar o <i>upload</i> do código-fonte	O usuário poderá associar a uma unidade concreta (que está associada a uma unidade abstrata) um arquivo contendo o código-fonte.
UC10	Executar o componente	A partir de uma instância de um componente (previamente selecionada ou a partir de um arquivo contendo a instância) o usuário poderá falar a resolução da instância e assim implantá-la em uma plataforma de execução.
UC11	Realizar o <i>download</i> das opções	Após a resolução da instância, o usuário poderá salvar o resultado em um arquivo.
UC12	Visualizar os componentes disponíveis	O sistema deve listar os componentes disponíveis organizados por espécies (tipos)
UC13	Criar componente de plataformas	O usuário poderá criar componentes de espécies específicas, com base no componente pai, neste caso, do tipo plataforma.
UC15	Criar Perfil de	O usuário poderá criar uma instância a partir

	plataforma	do componente da plataforma especificada pelo mesmo.
UC15	Manter a arquitetura de <i>workflows</i>	O usuário poderá criar e editar a arquitetura dos componentes que se unem para formar o <i>workflow</i> .
UC16	Criar fluxo do <i>workflow</i>	O usuário poderá criar e editar os fluxo de dados e procedimentos do <i>workflow</i> .
UC17	Gerenciar Preferências	O usuário poderá editar os valores que já foram padronizados segundo suas preferências
UC18	Criar perfis de plataforma	O usuário poderá criar instâncias componentes do tipo plataforma a partir do seu parque de recursos.

7.3. Diagrama de Casos de Uso



