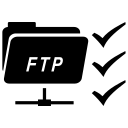
Ronaldo Pimentel Marques da Silva

FTP CHECKER

Pré-ProjetO

v 0.1.pt



São José do Rio Preto, SP, Brasil – 2022

1. INTRODUÇÃO

Este documento “pré-projeto” tem por finalidade dar ao leitor noções gerais dos assuntos preliminares à implementação de seu produto e de sua documentação. O público alvo deste pré-projeto é: seu “ARQUITETO DA SOLUÇÃO”, “GESTOR DO PROJETO” e “TIME DESENVOLVEDOR”, descriminados a frente neste documento no tópico “Partes Interessadas”.

O produto deste pré-projeto é um software verificador do estado de servidores *FTP* (*File Transfer Protocol* – Protocolo de Transferência de Arquivo), doravante denominado provisoriamente como “FTP CHECKER”.

FTP CHECKER visa ser um software *backend* (utilizado e executado no servidor *web* sem interação ou percepção do usuário de navegador *web*), iniciado e executado de forma análoga a um serviço de sistema operacional, com a finalidade de monitorar servidores *FTP* précadastrados quanto às suas disponibilidade de *ping*, disponibilidade de conexão, permissões de acesso às pastas e permissão de manipulação de arquivos, de forma a converter as informações coletadas no monitoramento em um repositório de informações a serem exibidas em uma página *web* dinâmica.

Este produto usa de sua metologia própria de desenvolvimento, a qual é influenciada pelos conceitos, métodos e ferramentas tais como Prototipação, *Agile*, *Scrum* entre outros. Portanto implementa o protótipo em caráter MVP (*Minimum Viable Product* – Mínimo Produto Viável) que possui menos funcionalidades que a versão final do produto. Questões estas descritas a frente neste documento nos tópico “Escopo do Produto” e “Metodologia”.

1.1. **Objetivo do Produto**

FTP CHECKER tem por objetivo gerar, manter, calcular, e emitir … // TODO:

1.**2.** ESCOPO DO PRODUTO

Considera-se escopo do FTP CHECKER:

* a existência, a conexão, o uso e manutenção de uma instância de banco de dados relacional com suas entidades e relacionamentos projetados para o FTP CHECKER;
* implantação, publicação acessível ao usuário através de um servidor *web* (e/ou servidor de aplicações), mesmo que de uso compartilhado (não dedicado);
* disponibilidade de acesso ao software e seus recursos conforme licença contratada;
* gestão (incluir, alterar, excluir) do cadastro de usuários, seus níveis e controle de acesso às funções do FTP CHECKER;
* gestão (incluir, alterar, excluir) do cadastro de insumos, composições de custos, BDI (benefícios e dispesas indiretas), leis sociais, parâmetros de curba abc e do orçamento propriamente dito;
* emição de relatórios em páginas *web* na inteface de usuário padrão do FTP CHECKER para que seja imprimida ou salva.

1.2.1. **Escopo do Protótipo MVP**

A… // TODO: a fazer este tópico. Escopo do MVP é menor...

1.**3.** ATUALIZAÇÃO DESTA DOCUMENTAÇÃO

Este pré-projeto é versionado em duas escalas, “versão princial” e “versão ajuste”, ambas representadas por números inteiro positivo.

A “versão principal” é incrementada devido às alterações de grande porte, ou seja, que alteram seu sentido, essência, objetivos, escopo do produto, escopo do projeto, ou requisitos. A “versão ajuste” é incrementada devido às alterações de pequeno-porte com finalidades corretiva no documento para melhor comunicação sem alterar sua interpretação ou seu resultado pretendido.

Por fim a “versão idioma” simboliza qual idioma no qual o documento foi elaborado. Sendo utilizado a padronização ISO 639-1, temos: “pt” para português, “es” para espanhol, “fr” para francês, e “em” para inglês.

A atualização deste documento somente ocorre pelo “ARQUITETO DA SOLUÇÃO” ou “GESTOR DO PROJETO”; No ato de atualização mantém-se o arquivo da versão anterior, e incrementa 1 ao número atual na escala que está sendo alterada.

Demonstração do versionamento:

[versão principal] **.** [versão ajuste] . [versão idioma]

Exemplos do versionamento:

1.0.pt , 1.1.es , 1.5.fr , 2.0.en , 2.9.pt

2. REQUISITOS

Neste projeto entende-se por requisito uma característica e/ou comportamento necessários e esperados no FTP CHECKER.

2.1. Classificação do Requisito

Este projeto classifica seus requisitos quanto à relevância e granularidade.

2.1.1. Relevância do Requisito

Este pré-projeto classisfica seus requisitos em três níveis de relevância: “Essencial”, “Importante” e “Desejável”, os quais podem ser expressos de forma abreviada em legenda identificadora de requisitos repectivamente por: “E”, “I” e “D”. O uso legenda identificadora de requisitos será mostrado a frente neste documento.

Os níveis de relevância de requisitos devem ser interpretado como segue:

* Essencial: requisito *sine qua non* (obrigatória), o FTP CHECKER não funciona, ou não há razão de existir sem tal requisito; é necessária para liberação da versão “MVP”; é necessário para liberação da versão final;
* Importante: necessária para cumprir o objetivo total a que se destina o FTP CHECKER; ausência do requisito pode acarretar limitação no funcionamento; não é necessária para liberação da versão “MVP”; mas é necessário para liberação da versão final;
* Desejável: aprimora a experiência ou eficiência no uso do produto; o produto funciona sem o requisito; não é necessária para liberação da versão “MVP”; não é necessário para liberação da versão final;

2.1.2. Granularidade do Requisito

Este projeto classisfica seus requisitos em quatro níveis de granularidade: “Épico”, “Grossa”, “Média” e “Fina”, os quais podem ser expressos de forma abreviada em um identificador de requisitos… // FIXME: (ALTERAR PARA 0.0.0.0) repectivamente por: “P”, “G”, “M” e “F”. O uso legenda identificadora de requisitos será mostrado a frente neste documento.

Os níveis de granularidade de requisitos devem ser interpretado como segue:

* Épico: requisitos elicitados em perspectiva abrangente e não profunda, numa visão superficial do produto, a qual traz os poucos requisitos primordiais com alto nível de abstrações, ou seja os requisitos mais perceptíveis do produto, cuja possuem outros requisitos e sub-requisitos internamente; // TODO: n.0.0.0
* Grossa: requisitos mais específicos que os de granularidade “épico” porém ainda possui nível relativamente alto de abstração das funções do produtom, ; // TODO: n.n.0.0
* Média: f...; // TODO: n.n.n.0
* Fina: f...; // TODO: n.n.n.n

2.2. IDENTIFICADOR DE Requisitos

Este projeto identifica seus requisitos com um acrônimo composto por letras e números, sendo “RF-n.n.n.n-r” para requisitos funcionais, e “RNF-n.n.n.n-r” para requisitos não funcionais, onde cada conjunto de caracteres deve ser interpretado da seguinte forma:

* “RF” : Requisito Funcional;
* “RNF” : Requisito Não Funcional;
* “n” : número inteiro positivo sequencial por ordem natural de criação do requisito, iniciando em “1”;
* “n.n.n.n” : cadeia de “n” separadas por ponto representa a granularidade do requisito, sendo o primeiro “n” enumerado com valor positivo, e demais “n”s com valor “0” (zero) denota-se o primeiro nível de granularidade, “épico”; o segundo “n” enumerado com valor positivo, e demais “n”s à direita enumerados com valor “0” (zero) denota-se o segundo nível de granularidade, “grossa”; o terceiro “n” enumerado com valor positivo, e o último “n” à direita enumerados com valor “0” (zero) denota-se o terceiro nível de granularidade, “média”; e o último “n” à direita enumerados com valor positivo denota-se o quarto nível de granularidade, “fina”;
* “r” : letra maiúscula representante da relevância do requisito, citada no subtópico 2.1.1.

Exemplos:

|  |  |
| --- | --- |
| RF-1.0.0.0-E | Requisito Funcional, Épico “1”, Essencial; |
| RNF-1.0.0.0-E | Requisito Não Funcional, Épico “1”, Essencial; |
| RF-2.0.0.0-E | Requisito Funcional, Épico “2”, Essencial; |
| RNF-2.0.0.0-E | Requisito Não Funcional, Épico “2”, Essencial; |
| RF-1.1.0.0-E | Requisito Funcional, Épico “1”→Grossa “1” , Essencial; |
| RF-1.2.0.0-I | Requisito Funcional, Épico “1”→Grossa “2” , Importante; |
| RF-1.18.0.0-I | Requisito Funcional, Épico “1”→Grossa “18” , Importante; |
| RNF-2.3.28.5-D | Requisito Não Funcional, Épico “2”→ Grossa “3”→ Média “28”→ Fina “5” , Desejável. |

2.3. OS Requisitos Funcionais

* RF-1.0.0.0-E ...

2.4. os Requisitos Não Funcionais

* RNF-1.0.0.0-E Estar acessível atravé da internet em um endereço *web*;
* RNF-1.1.0.0-I Restringir acesso e uso exclsivamente à usuários précadastrados, sendo cadastro individual por pessoa física, mesmo que atrelado a um contrato de licença de pessoa jurídica;
* RNF-1.1.1.0-I Gerir controle de acesso do usuário vinculando-o com número de CPF, e-mail e senha;
* RNF-1.1.1.1-I Número de CPF deve ser validado, aceitando-se somente números de CPF válidos (existentes);
* RNF-2.0.0.0-E Ser perfeitamente utilizável conforme projetado, nos sistemas operacionais Microsoft Windows 7 ou superior, através dos navegadores *web* *Brave Browser* v1.18 ou superior, e *Google Chrome* vX.Y ou superior;
* RNF-2.1.0.0-I Ter interface de usuário responsiva em suas dimensões, para os navegadores *web* citados em RNF-2.0.0.0-E;
* RNF-2.2.0.0-I Não possibilitar usuário autenticar-se e/ou utilizar duas instâncias simutaneamente;
* RNF-2.2.1.0-I Na tentativa do usuário autenticar-se em duas instâncias simutaneamente, deve-se questioná-lo de desconectar a instância (anterior) até então em uso para iniciar a nova instância com a tentativa de autenticação;
* // HERE: continuar...

2.5. Requisitos Não Contemplados (escopo negativo)

A…

3. **METODOLOGIA**

Há diversas metodologia de desenvolvido de porojetos no mercado da indústria de softwares. Cascata, Prototipação e Agile são algumas das mais utilizadas e estudadas nas últimas décadas.

A fim de otimizar os resultados o projeto FTP CHECKER utiliza-se de um método próprio, influenciado pelos três métodos a cima citados, acrescentado do uso parcial do *framework Scrum* e *Kanban*.

3.1. **Protótipo MVP**

A.

4. **partes interessadas**

Comumente também denominado stakeholders

Cliente do projeto:

Usuário:

Gerente do projeto:

4.1. **A...**

A…

**5.** **CONSIDERAÇÕES FINAIS...**

A…

**5**.1. **A...**

A…