

netMap

Documento de Requisitos

2019

PREFÁCIO

Versão	Data	Autor	Razão
0.1	13/09/2019	Giovane de Loredo	Criação do documento

GLOSSÁRIO

JavaScript: Linguagem de programação interpretada de alto nível, dinâmica, fracamente tipificada, prototype-based e multi-paradigma. Juntamente com HTML e CSS, o JavaScript é uma das três principais tecnologias da World Wide Web.

Node.js: Interpretador de código JavaScript que tem como principais características o fluxo de trabalho assíncrono e orientado a eventos. É conhecido por migrar a programação com Javascript do lado do cliente para os servidores.

Express: Framework para aplicativo da web do Node.js mínimo e flexível que fornece um conjunto robusto de recursos para aplicativos web e móvel.

React: Biblioteca JavaScript de código aberto para criar interfaces de usuário.

Material-UI: Biblioteca de componentes React construídos com base na documentação do Material Design do Google.

MongoDB: Software de banco de dados orientado a documentos livre e multiplataforma. Classificado como um programa de banco de dados NoSQL, o MongoDB usa documentos semelhantes a JSON com esquemas.

Redis: É um armazenamento de estrutura de dados de chave-valor de código aberto e na memória.

Docker: É uma tecnologia de software que fornece contêineres. Fornece uma camada adicional de abstração e automação de virtualização de nível de sistema operacional no Windows e no Linux.

SUMÁRIO

INTRODUÇÃO	4
ARQUITETURA DO SISTEMA	5
REQUISITOS NÃO FUNCIONAIS DE USUÁRIO	6
REQUISITOS FUNCIONAIS DE USUÁRIO	7
REQUISITOS NÃO FUNCIONAIS DE SISTEMA	8
REQUISITOS FUNCIONAIS DE SISTEMA	9

INTRODUÇÃO

O netMap é um *software* web desenvolvido com o intuito de facilitar a organização e padronização das redes de computadores entre os campi do IFSP. Permite ao responsável pela manutenção das redes de computadores dos campi estabelecer padrões para criação e configuração das VLANs, gerando uma fácil comunicação com os responsáveis por manter a rede de computador de cada campus.

Motivação

Devido a cada campus do IFSP possuir sua rede de computadores configurada de forma diferente, dificulta-se muito o trabalho do responsável por manter o pleno funcionamento de todos os campi. Como tentativa de solucionar o problema, foi desenvolvida uma tabela que fornece uma padronização a ser seguida por todos os campi para definir a estrutura da rede de computadores. Este trabalho resume-se na criação de um software que busca reproduzir o sucesso alcançado com a atual ferramenta utilizada - a tabela - agregando mais eficiência e segurança ao processo.

Resumo do sistema

Nome: netMap

Tipo: Aplicação Web

Linguagem de programação: JavaScript

Biblioteca de interface do usuário: React e Material-UI

Framework: Express

Banco de dados: MongoDB e Redis

ARQUITETURA DO SISTEMA

O sistema desenvolvido utiliza o padrão Model View Controller (MVC) para organizar suas estruturas internas, tendo todas as suas camadas escritas em uma mesma linguagem de programação, o JavaScript. São utilizadas três camadas:

- **Aplicação (View):** Camada responsável pela interface com o usuário. É construído de maneira modular através da componentização. Utiliza a biblioteca de componentes prontos Material-UI, que é desenvolvida com a biblioteca para construção de interface do usuário React.
- **Controle (Controller):** Camada responsável pela comunicação entre a camada de aplicação e o banco de dados. Nessa camada são implementadas as regras de negócio do sistema.
- **Modelo (Model):** Define o esquema que deve ser salvo como documento no MongoDB, realizando também todas as validações de tipos de dado necessárias.

REQUISITOS NÃO FUNCIONAIS DE USUÁRIO

Descreve de forma concisa as características gerais que o sistema necessita possuir.

RNF01 - Usabilidade

O sistema deve ter uma interface do usuário amigável e notificar o usuário do sucesso ou fracasso de uma operação.

RNF02 - Segurança

O sistema deve armazenar os dados de forma segura, validando todas as entradas e utilizando criptografia quando necessário.

RNF03 - Multiplataforma

O sistema deve funcionar em qualquer navegador de internet moderno.

REQUISITOS FUNCIONAIS DE USUÁRIO

Descreve as funcionalidades do sistema de maneira simples e concisa.

RF01 - Primeiro acesso

O sistema deve exibir um formulário para cadastro do primeiro usuário do sistema, quando o sistema for iniciado pela primeira vez.

RF02 - Autenticação

O sistema deve exigir informações para autenticar o usuário permitindo o uso do sistema como um todo.

RF03 - Recuperação de senha

O sistema deve possibilitar a alteração da senha em caso de perda.

REQUISITOS NÃO FUNCIONAIS DE SISTEMA

Descreve as propriedades gerais do sistema de forma completa, indicando características e métricas utilizadas, quando necessário.

RNF01 - Usabilidade

Desenvolvido com a biblioteca de componentes Material-UI, é garantido que o sistema possua uma sólida coleção de componentes testados e padronizados, assim como um sistema de tema e layout integrado. Seguindo os princípios de design do Material Design, a documentação para construção de interfaces do usuário do Google, é garantido que o sistema seja amigável e esteja de acordo com as melhores práticas atuais.

RNF02 - Segurança

Todas as tecnologias utilizadas na construção do sistema são gratuitas, de código aberto e possuem alta adoção por parte da comunidade. As informações são armazenadas no MongoDB, um sistema de banco de dados não relacional, o que implica em não possuir esquemas que definem quais dados e seus tipos devem estar ali contidos, entretanto, todas as entradas são validadas por um poderoso par de ferramentas: Joi e Mongoose.

RNF03 - Multiplataforma

Com a camada de aplicação sendo construída com o React, a biblioteca para construção de interfaces do usuário mais utilizada do mercado e mantida pelo gigante Facebook, o sistema garante alta performance e compatibilidade com todos os navegadores de internet modernos.

REQUISITOS FUNCIONAIS DE SISTEMA

Descreve as funcionalidades do sistema de maneira completa, apresentando todos os detalhes necessários para implementação e/ou entendimento do fluxo de trabalho dentro do sistema.

RF01 - Primeiro acesso	
Função	Permite o cadastro do primeiro usuário do sistema.
Descrição	Insere um novo usuário na coleção Users contendo as informações fornecidas no formulário de cadastro.
Entradas	Nome (String), Prontuário (String), E-mail (String) e Senha (String). Todos os campos são obrigatórios.
Origem	Não se aplica.
Saída	Sucesso ou falha na criação do usuário.
Destino	Tela de autenticação.
Ação	Caso seja a primeira vez que o sistema esteja sendo acessado, nenhum usuário existe no banco de dados, desta forma, o usuário é redirecionado para a tela de cadastro do primeiro usuário.
Pré-condição	O sistema e todas as suas dependências deve estar em execução.
Pós-condição	O sistema notifica o usuário do sucesso ou falha da operação.
Efeitos colaterais	Nenhum.

RF02 - Autenticação	
Função	Permite a autenticação do usuário no sistema.
Descrição	Cria uma sessão com base nas informações fornecidas.
Entradas	Prontuário (String) e Senha (String). Todos os campos são obrigatórios.
Origem	Não se aplica.
Saída	Sucesso ou falha na autenticação.

Destino	Tela principal.
Ação	Ao solicitar ao navegador web para acessar o endereço do sistema, é exibida uma tela solicitando as informações de autenticação.
Pré-condição	O sistema e todas as suas dependências deve estar em execução.
Pós-condição	O sistema notifica o usuário do sucesso ou falha da operação.
Efeitos colaterais	Nenhum.

RF03 - Recuperação de senha	
Função	Permite a alteração da senha em caso de perda.
Descrição	Envia um e-mail para o usuário que redireciona para uma página de alteração de senha.
Entradas	E-mail (String). Todos os campos são obrigatórios.
Origem	Tela de autenticação.
Saída	Sucesso ou falha no envio do e-mail.
Destino	Nenhum.
Ação	Ao solicitar ao navegador web para acessar o endereço do sistema, é exibida uma tela solicitando as informações de autenticação. Nesta tela existe a opção “Esqueceu a senha?”, que quando selecionada redireciona o usuário para tela de solicitação do e-mail.
Pré-condição	O sistema e todas as suas dependências deve estar em execução.
Pós-condição	O sistema notifica o usuário do sucesso ou falha da operação.
Efeitos colaterais	Nenhum.