

PETI - Planejamento Estratégico de Tecnologia da Informação

1. Objetivo

Contando com visitas de noventa por cento dos usuários da internet brasileira a disponibilidades dos produtos do UOL é uma preocupação constante. Foi observado que bons processos de monitoramento de aplicações nos auxiliam na criação de soluções resilientes e melhoram a reputação da marca. Sabendo disso, será desenvolvido um software com alternativas disruptivas que alertam sobre os riscos de indisponibilidade dos produtos UOL.

2. Escopo

1. Sprint 1

1.1. Formulário

- 1.1.1.Criação das rotas
- 1.1.2.Criação do banco
- 1.1.3.Tela de cadastro
- 1.1.4.Tela de listagem

1.2. Envio de e-mails

- 1.2.1.Criação do banco mongoDB
- 1.2.2.Implementação do serviço Prometheus
- 1.2.3.Implementação do servidor NiFi
- 1.2.4.Automatização do processo de coleta e tratamento
- 1.2.5.Implementação de serviço de e-mail
- 1.2.6.Implementação do serviço Firebase
- 1.2.7.Implementar servidor Flask
- 1.2.8.Gerar JSON de dados do Prometheus
- 1.2.9.Gerar CSV a partir dos dados do JSON
- 1.2.10. Implementar funções que retornam dados de consumo de um intervalo de período, média de consumo, níveis de consumo, picos de uso e crescimento em relação ao período anterior (CPU, RAM, Disco, número e tempo de acesso)
- 1.2.11. Implementar alerta de ultrapassagem de consumo de CPU, RAM, Disco, número e tempo de acesso (a partir do valor definido)
- 1.2.12. Implementar disparo de e-mail periódico

1.3. Dashboard auxiliar

- 1.3.1.Implementar interface gráfica do dashboard (CPU, RAM, Disco, Número e tempo de requisições em um período)

2. Sprint 2

2.1. Menu de Configuração

- 2.2. Gráfico com previsão de falha CPU e RAM
- 2.3. Visualizar Gráfico (Expandir imagem)
- 2.4. Indicador de previsão de falha
- 2.5. Coleta e classificação dos dados de tempo de resposta
- 2.6. Coleta e classificação de dados de número de requisições
- 2.7. Criação do modelo - Treinamento diário (scheduler) (global)
- 2.8. Função que retorna dados de previsão (período)
- 2.9. Função que analisa periodicamente possibilidades de falha
- 2.10. Função que cria tarefa (scheduler) email periódico
- 2.11. Acrescentar informações de previsão ao email (periódico)
- 2.12. Função que altera scheduler existente - email periódico
- 2.13. Subir NiFi

- 2.14. Subir Dashboard-api
 - 2.15. Subir form-client
 - 2.16. Subir Prometheus
 - 2.17. Subir dashboard-client
 - 2.18. Subir Spring
 - 3. Sprint 3
 - 3.1. Previsão de consumo de tráfego de rede
 - 3.1.1. Disparo de alerta periódico com níveis de criticidade
 - 4. Feira de Soluções
 - 4.1. Apresentação do projeto
3. Metodologia
- Foram utilizadas metodologias ágeis (em nosso caso o SCRUM) no desenvolvimento deste projeto, de forma que o processo de construção seja rápido, prático e eficaz. A intenção foi tornar o desenvolvimento em algo flexível e iterativo.
4. Mapa Estratégico

