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.

1. Escopo

1. Sprint 1

1.1. Formulário

* + 1. Criação das rotas
    2. Criação do banco
    3. Tela de cadastro
    4. Tela de listagem

1.2. Envio de e-mails

* + 1. Criação do banco mongoDB
    2. Implementação do serviço Prometheus
    3. Implementação do servidor NiFi
    4. Automatização do processo de coleta e tratamento
    5. Implementação de serviço de e-mail
    6. Implementação do serviço Firebase
    7. Implementar servidor Flask
    8. Gerar JSON de dados do Prometheus
    9. Gerar CSV a partir dos dados do JSON
    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)
    11. Implementar alerta de ultrapassagem de consumo de CPU, RAM, Disco, número e tempo de acesso (a partir do valor definido)
    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

* 1. Menu de Configuração
  2. Gráfico com previsão de falha CPU e RAM
  3. Visualizar Gráfico (Expandir imagem)
  4. Indicador de previsão de falha
  5. Coleta e classificação dos dados de tempo de resposta
  6. Coleta e classificação de dados de número de requisições
  7. Criação do modelo - Treinamento diário (scheduler) (global)
  8. Função que retorna dados de previsão (período)
  9. Função que analisa periodicamente possibilidades de falha
  10. Função que cria tarefa (scheduler) email periódico
  11. Acrescentar informações de previsão ao email (periódico)
  12. Função que altera scheduler existente - email periódico 2.13. Subir Nifi
  13. Subir Dashboard-api
  14. Subir form-client
  15. Subir Prometheus
  16. Subir dashboard-client
  17. Subir Spring

1. Sprint 3
   1. Refatoração do locust loader
   2. Deploy das aplicações form-server; locust-loader; banco postgres e flask-etl no Azure
   3. Aprimoramento do modelo de machine learning
   4. Implantação do modelo de machine learning como microserviço
   5. Implantação das métricas de saúde e previsão de falha no dashboard
   6. Adição da previsão de saúde no e-mail periódico
   7. Deploy do machine learning
2. Feira de Soluções
   1. Apresentação do projeto

1. 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.

1. Mapa Estratégico

