|  |  |  |  |
| --- | --- | --- | --- |
| **Controle de Versões** | | | |
| **Versão** | **Data** | **Autor** | **Notas da Revisão** |
| 1.0 | 29/03/2022 | Time Pycemaker | Elaboração Inicial – Protótipo Formulário |

# Objetivos deste documento

Descrever de forma clara qual trabalho deverá ser realizado e quais entregas serão produzidas.

# Objetivos do projeto

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.

# Escopo do Produto

Requisitos Funcionais:

* Desenvolver um formulário de cadastro, com os campos: Nome, E-mail, Senha e Celular.
* Desenvolver um formulário de consulta dos cadastros realizados.
* Prever quando ocorrerá uma falha de software que causará a indisponibilidade da Aplicação.

Requisitos Não Funcionais:

* O tempo de resposta do backend de cadastros deve ser abaixo de 300ms.
* O tempo de resposta do backend de consulta de cadastros deve ser abaixo de 100ms.

# Exclusões do projeto / Fora do Escopo

Serão consideradas atividades fora do escopo, qualquer outra atividade que não esteja reportada no tópico acima.

# Restrições

Orçamento total do projeto: R$ 50.000,00;

Data de finalização do projeto: 05/06/2022;

Não exceder o valor de R$2000,00 por membro da equipe para cada sprint;

O projeto não pode exceder o tempo de duração de 3 meses.

# Premissas

Os dados devem estar íntegros para poderem ser analisados;

O tempo de resposta da aplicação não deve ultrapassar 300ms (para o cadastro) e 100ms (para consulta).

# Estrutura Analítica do Projeto

1. Sprint 1
   1. Formulário
      1. Criação das rotas
      2. Criação do banco
      3. Tela de cadastro
      4. Tela de listagem
   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
   3. Dashboard auxiliar
      1. Implementar interface gráfica do dashboard (CPU, RAM, Disco, Número e tempo de requisições em um período de tempo)
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 nova tarefa (scheduler) email periódico
   11. Acrescentar informações de previsão ao email (periódico)
   12. Função que altera scheduler existente - email periódico
   13. Subir Nifi
   14. Subir Dashboard-api
   15. Subir form-client
   16. Subir Prometheus
   17. Subir dashboard-client
   18. Subir Spring
3. 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
4. Feira de Soluções
   1. Apresentação do projeto

# Critérios de Aceitação

[Defina os critérios de aceitação das entregas definidas na WBS e do projeto como um todo]

|  |  |  |
| --- | --- | --- |
| **Cod. EAP** | **Entrega** | **Critérios de aceitação** |
| 1 | Sprint 1 | Atender as exigências do cliente dentro das restrições estipuladas. |
| 1.1 | Formulário | Implementar formulário em SpringBoot e Frontend SPA com Vue.js |
| 1.1.1 | Criação das rotas | Deve retornar as rotas de Cadastro e Listagem. |
| 1.1.2 | Criação do banco | Armazenar Cadastro dos Usuários |
| 1.1.3 | Tela de cadastro | Uma Tela onde o usuário fará o seu Cadastro. |
| 1.1.4 | Tela de listagem | Tela que lista os usuários. |
| 1.2 | Envio de e-mails | Criação do banco para armazenar as métricas, envio de alertas por e-mail (Grafana e IA), dashboard de monitoramento. |
| 1.2.1 | Criação do banco mongoDB | O banco deve conseguir receber todas as métricas |
| 1.2.2 | Implementação do serviço Prometheus | O serviço deve estar implementado |
| 1.2.3 | Implementação do servidor NiFi | O servidor deve estar implementado |
| 1.2.4 | Automatização do processo de coleta e tratamento | Deve coletar, tratar e armazenar os dados no bando mongodb |
| 1.2.5 | Implementação de serviço de e-mail | Os e-mails devem estar sendo enviados |
| 1.2.6 | Implementação do serviço Firebase | O serviço deve estar rodando para armazenar as imagens |
| 1.2.7 | Implementar servidor Flask | O servidor deve estar implementado |
| 1.2.8 | Gerar JSON de dados do Prometheus | Os dados devem estar sendo gerados corretamente |
| 1.2.9 | Gerar CSV a partir dos dados do JSON | Os dados devem estar sendo gerados corretamente |
| 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) | Todas as funções devem retornar valores reais e corretos |
| 1.2.11 | Implementar alerta de ultrapassagem de consumo de CPU, RAM, Disco, número e tempo de acesso (a partir do valor definido) | O alerta deve estar sincronizado com o tempo real |
| 1.2.12 | Implementar disparo de e-mail periódico | O disparo deve estar sendo realizado corretamente |
| 1.3 | Dashboard auxiliar | Exibe em tempo real o consumo de Hardware e Requisições |
| 1.3.1 | Implementar interface gráfica do dashboard (CPU, RAM, Disco, Número e tempo de requisições em um período de tempo) | A interface deve estar apresentável e funcionando |
| 2 | Sprint 2 |  |
| 2.1 | Menu de Configuração | O layout deve estar finalizado e funcional |
| 2.2 | Gráfico com previsão de falha CPU e RAM | O layout deve estar finalizado e funcional |
| 2.3 | Visualizar Gráfico (Expandir imagem) | O layout deve estar finalizado e funcional |
| 2.4 | Indicador de previsão de falha | O layout deve estar finalizado e funcional |
| 2.6 | Coleta e classificação dos dados de tempo de resposta | A operação deve ter sido efetiva |
| 2.7 | Coleta e classificação de dados de número de requisições | A operação deve ter sido efetiva |
| 2.8 | Criação do modelo - Treinamento diário (scheduler) (global) | O modelo deve estar funcional |
| 2.9 | Função que retorna dados de previsão (período) | A função deve retornar os dados |
| 2.10 | Função que analisa periodicamente possibilidades de falha | A função deve analisar corretamente as possibilidades de falha |
| 2.11 | Função que cria tarefa (scheduler) email periódico | A função deve retornar o scheduler |
| 2.12 | Acrescentar informações de previsão ao email (periódico) | As informações devem constar no email |
| 2.13 | Função que altera scheduler existente - email periódico | A função deve estar ativa |
| 2.14 | Subir Nifi | O deploy da aplicação deve estar funcionando |
| 2.15 | Subir Dashboard-api | O deploy da aplicação deve estar funcionando |
| 2.16 | Subir form-client | O deploy da aplicação deve estar funcionando |
| 2.17 | Subir Prometheus | O deploy da aplicação deve estar funcionando |
| 2.18 | Subir dashboard-client | O deploy da aplicação deve estar funcionando |
| 2.19 | Subir Spring | O deploy da aplicação deve estar funcionando |
| 3 | Sprint 3 |  |
| 3.1 | Previsão de consumo de tráfego de rede | Aprimoramento do dashboard, e-mails de alerta |
| 3.1.1 | Disparo de alerta periódico com níveis de criticidade | Dispara um alerta via e-mail de periodicamente em relação ao consumo de Hardware. |
| 4 | Feira de Soluções |  |
| 4.1 | Apresentação do projeto |  |

|  |  |  |
| --- | --- | --- |
| **Aprovações** | | |
| **Participante** | **Assinatura** | **Data** |
| Patrocinador do Projeto |  |  |
| Gerente do Projeto |  |  |