

	Declaração de Escopo	
	Sistema de Prevenção de Erro	

Controle de Versões			
Versão	Data	Autor	Notas da Revisão
1.0	23/03/2022	Rodrigo Hideaki Ando	Elaboração Inicial – primeiro rascunho

Objetivos deste documento

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

O presente documento estabelece parâmetros para realizar o Projeto Integrador. A sequência de eventos será estabelecer uma métrica de utilização de recursos, a utilização de recursos em ambiente controlado, a capacidade máxima de recursos e a resposta dos serviços frente a um evento inesperado.

Objetivos do projeto

O Trabalho é mais intelectual do que prático, apesar de ter algo prático. Teremos que focar nas técnicas e métricas para a redução de variáveis para a alocação de recursos para a disponibilidade de serviços no tempo corrente e promover sua continuidade no futuro, controlando incertezas. Uma das estratégias será não trabalhar com todas as variáveis, mas algumas e focar na disponibilidade daquele recurso. Serão analisados dados de monitoramento de rede, será quantificada a porcentagem de utilização das classes de recursos, sem a utilização máxima do hardware e será estabelecido um ponto ótimo entre eficiência e confiabilidade.

Escopo do Produto

Após a reunião com o cliente, houve uma discussão com a equipe e definiu-se que a prevenção de erro é o ponto chave do projeto. Contudo, será necessário desenvolver um sistema de cadastro para ser usado como aplicação alvo do sistema principal responsável por monitorar e prever indisponibilidades.

Vide detalhes na EAP e nos respectivos critérios de aceitação.

Exclusões do projeto / Fora do Escopo

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

Restrições

Data de limite para finalização do projeto: 05/06/2022. O Projeto não pode ultrapassar o tempo de 90 dias.

O limite de custo para cada membro da equipe deve ser de 2.000,00 R\$ por Sprint.

Premissas

Para cumprir o prazo determinado, os membros da equipe, em consenso, exigiram como premissa:

- A cada membro da equipe, relatar o andamento de sua respectiva tarefa durante a semana.
- A equipe deve realizar uma reunião semanal para definir novas tarefas e demonstrar os resultados obtidos.
- Atualizar semanalmente as ferramentas de gerenciamento utilizadas no projeto, como, por exemplo, o Jira e o GitHub.
- Por Sprint, a equipe deve apresentar o andamento do projeto ao cliente.
- Consultar semanalmente os professores responsáveis por coordenar o projeto.

Estrutura Analítica do Projeto

1. Definição do Projeto
2. Gerenciamento do Projeto
 - 2.1 Design Thinking
 - 2.2 Redigir Documentos
 - 2.2.1 Termo de abertura do projeto
 - 2.2.2 Gerenciamento de Escopo do Projeto
 - 2.2.3 Gerenciamento de Cronograma
 - 2.2.4 Gerenciamento de Custo
 - 2.2.5 Gerenciamento de Qualidade
 - 2.3 Ferramentas de Gerenciamento
 - 2.3.1 Definir Backlog
 - 2.3.2 Criar repositório
3. Sistema de Cadastro
 - 3.1 Backend
 - 3.1.1 Criar Classes
 - 3.1.2 Relacionar as Classes
 - 3.1.3 Preparar o banco de dados
 - 3.2 Frontend
 - 3.2.1 Criar protótipo das telas
 - 3.2.2 Desenvolver a interface
 - 3.2.3 Conectar com backend
4. Sistema de Prevenção de Erro
 - 4.1 Backend
 - 4.1.1 Relacionar com o sistema alvo
 - 4.1.2 Identificar as métricas
 - 4.1.3 Preparar o banco de dados
 - 4.1.4 Treinar IA
 - 4.1.5 Receber e Tratar dados
 - 4.1.6 Prever e Notificar os erros
 - 4.2 Frontend
 - 4.2.1 Criar protótipo das telas

- 4.2.2 Desenvolver interface
- 4.2.3 Conectar com backend
- 4.2.4 Dashboard

5. Testes

- 5.1 Testar Sistema de cadastro
 - 5.1.1 Enviar e consultar dados de cadastro
- 5.2 Testar Sistema de prevenção de erro
 - 5.2.1 Induzir a aplicação alvo ao erro
 - 5.2.2 Funcionamento da prevenção de erro
 - 5.2.3 Coerência de dados
- 5.3 Teste Final

Critérios de Aceitação

Cod. EAP	Entrega	Critérios de aceitação
1	Definição do Projeto	Atender as exigências do Cliente da UOL e dos professores responsáveis por coordenar o projeto.
2	Gerenciamento do Projeto	
2.1	Design Thinking	Atender as exigências do professor Claudio Etelvino de Lima, responsável por lecionar a disciplina Gestão de Governança de TI.
2.2	Redigir Documentos	Atender as exigências do professor Claudio Etelvino de Lima, responsável por lecionar a disciplina Gestão de Projetos.
2.3	Ferramentas de Gerenciamento	Vide detalhamento abaixo.
2.3.1	Definir Backlog	Utilizando da ferramenta Jira, definir o backlog das três sprints do projeto. Extrair o burndown através dela.
2.3.2	Criar Repositório	Criar um repositório no GitHub. Buscar alinhar os participantes sobre o andamento do projeto.

3	Sistema de Cadastro	
3.1	Backend	Vide detalhamento abaixo.
3.1.1	Criar Classes	Criar as classes utilizando Spring Boot.
3.1.3	Criar Banco de dados	Criar as entidades no banco de dados e relacioná-los corretamente.
3.2	Frontend	Vide detalhamento abaixo.
3.2.1	Criar protótipo das telas	Fazer um protótipo da interface de cadastro de usuário e da tabela de usuário no Figma.
3.2.2	Desenvolver a interface	Desenvolver a interface seguindo o protótipo. Utilizar o Vue.js e CSS.
3.2.3	Conectar com o backend	Utilizar dos endpoints para realizar a comunicação do frontend com backend.
4	Sistema de Prevenção de Erro	
4.1	Backend	Vide detalhamento abaixo.
4.1.1	Relacionar com o sistema alvo	Usar Netdata (Prometheus e Graphana) para receber os dados da aplicação alvo.
4.1.2	Monitorar as métricas	Identificar e monitorar as métricas da aplicação alvo que podem indicar uma futura falha no sistema.
4.1.3	Preparar o banco de dados	Preparar o banco para armazenar os dados da aplicação alvo
4.1.4	Treinar IA	Treinar inteligência artificial para monitorar as métricas da aplicação alvo e identificar anormalidades no sistema.
4.1.5	Receber e Tratar dados	Receber e tratar os dados fornecidos pela IA. Gerar relatórios utilizando os dados recebidos
4.1.6	Prever e Notificar os erros	Desenvolver, com todos os dados adquiridos, um algoritmo capaz de prever e notificar as anormalidades do sistema, podendo também apresentar métodos de resolução de acordo com o problema apresentado.
4.2	Frontend	Vide detalhamento abaixo.

4.2.1	Criar protótipo das telas	Fazer um protótipo, utilizando o Figma, da interface responsável por apresentar os dados da aplicação alvo ao usuário.
4.2.2	Desenvolver a interface	Desenvolver a interface seguindo o protótipo. Utilizar o Vue.js e CSS.
4.2.3	Conectar com backend	Utilizar dos endpoints para realizar a comunicação do frontend com backend.
4.2.4	Dashboard	Incluir um dashboard com os dados recebidos do backend. Utilizar Bootstrap.
5	Testes	
5.1	Testar Sistema de Cadastro	Verificar se o sistema de cadastro está registrando e resgatando os dados no banco a partir dos formulários disponíveis na interface da aplicação.
5.2	Testar sistema de prevenção de erro	Vide detalhamento abaixo.
5.2.1	Induzir aplicação alvo ao erro	Induzir os problemas na aplicação alvo para testar as funcionalidades do sistema de prevenção de erro.
5.2.2	Funcionalidade da prevenção de erro	Verificar se o sistema é capaz de identificar e notificar os problemas induzidos intencionalmente na aplicação alvo.
5.2.3	Coerência de dados	Verificar a congruência dos dados apresentados pelo sistema.
5.3	Teste Final	Realizar uma última etapa de teste do sistema completo para realizar a conclusão do projeto.

Aprovações		
Participante	Assinatura	Data
Patrocinador do Projeto		
Gerente do Projeto		

--	--	--