

Requisitos do projeto Web – Princípios de Engenharia de Software

Luiz Carlos R. Viana - 1810936

11 de Abril de 2020.

Introdução

O objetivo deste projeto é construir um software web que poderá ser acessado por navegador, para o cumprimento das exigências de avaliação da disciplina INF1629 (Princípios de Engenharia de Software), **na sua nota de G2**. Conforme já explicitado no primeiro documento de referência do trabalho 2, a entrega do resultado final será no dia 2 de Julho, depois de 12 aulas dedicadas ao desenvolvimento do produto, cujas reuniões serão feitas separadamente para cada grupo. O trabalho deverá necessariamente ser hospedado no **GitHub**, e ali desenvolvido.

A nota do projeto será distribuída através de 4 pontos de controle, sendo as datas de cada ponto respectivamente: 14 de Maio, 26 de Maio, 4 de Junho, e 23 de Junho. Cada ponto de controle terá o valor de 1.5 pontos, e a entrega final valerá 4 pontos; somando 10 pontos no total. A avaliação será feita com base na qualidade do que for apresentado a cada ponto de controle, sendo esta julgada sobretudo pelo **uso, no trabalho, do que foi visto durante a primeira parte do curso**.

Admitiu-se neste trabalho a possibilidade de que cada grupo de alunos escolhesse o próprio tema do projeto, e as tecnologias a serem utilizadas para atingir a finalidade proposta. Deste modo, elegemos que o site a ser desenvolvido será uma aplicação voltada a **inspetores profissionais de artefatos** de projeto de software, especialmente de documentos de requisitos. Por meio deste site um inspetor será capaz de registrar suas próprias listas de inspeção e

documentos a serem inspecionados em uma conta própria particular, e poderá aplicar cada lista a um destes documentos, mantendo um histórico de inspeções. O grande modelo de nossa aplicação é o website **GitHub**, de modo que seria ideal que produzíssemos um sistema simples de upload e versionamento de repositórios de documentos que siga o padrão deste site. Devido a isto escolhemos denominar nossa aplicação **RevisionHub**.

Para tal fim, grande foi nosso aprendizado na leitura de (Kalinowski et al.) a respeito do processo de inspeção. Neste paper o autor descreve o processo de inspeção instaurado por (Fagan, 1976), visando a redução de erros no desenvolvimento de programas, descrevendo passo a passo o processo de inspeção em cascata intencionado por Fagan, e todos os seus atores.

Sumário da Arquitetura

Nós, do grupo 1 (G1), deliberamos que para a consecução de tal projeto, o *framework* **express.js** seria o mais indicado para a parte de *backend* do site, sendo esta biblioteca responsável por auxiliar no desenvolvimento do servidor de aplicação. Ao mesmo tempo, elegemos o *framework* **React.js**, desenvolvido pelo **FaceBook**, para o desenvolvimento *frontend*, devido à sua capacidade de produzir páginas web responsivas e reativas, à sua utilização de técnicas abstratas de programação funcional, e à sua considerável adoção no mercado. Para a persistência de dados utilizamos o sistema de banco de dados **No-Sql MongoDB**.

Requisitos

Requisitos para Usuários

[RU1]: O usuário logado deve ser capaz de fazer upload de listas públicas ou privadas de inspeção (checklists).

[RU2]: O usuário logado deve ser capaz de fazer upload de documentos públicos ou privados para serem inspecionados.

[RU3]: O usuário deve ser capaz de logar no sistema e gerenciar (C.R.U.D) suas listas de inspeção e documentos a serem inspecionados.

[RU4]: O usuário logado poderá editar suas listas de inspeção por um editor especial.

[RU5]: O usuário deve ser capaz de inspecionar privadamente um documento privado baseado em uma lista de inspeção privada.

[RU6]: O usuário deve ser capaz de acessar um histórico de suas inspeções.

[RU7]: O usuário deve ser capaz de buscar listas de inspeção pelo nome e datas de criação e de última modificação.

Requisitos para Transeuntes

[RT1]: Qualquer internauta, sem precisar estar logado, poderá acessar listas públicas de inspeção.

[RT2]: Qualquer internauta, sem precisar estar logado, poderá acessar documentos públicos a serem inspecionados.

[RT3]: Qualquer internauta poderá ranquear documentos e listas por inspeção em uma escala de 1 a 5 (usando estrelas).

[RT4]: Qualquer internauta, sem precisar estar logado, poderá inspecionar um documento público com uma lista de inspeção pública, e baixar os resultados da inspeção para sua máquina.

[RT5]: Nenhum internauta poderá editar o conteúdo de qualquer lista de inspeção ou documento sem estar logado.

Requisitos Não-Funcionais (Qualis)

[RSEG]: Os formulários de upload de listas e documentos do sistema serão resistentes a ataques do tipo SQL injection, etc... (Requisito de segurança)

Requisitos de Projeto

[RP1]: O código do projeto deve estar versionado no **GitHub**.

Casos de uso

UCU1 - Gerenciamento de checklists

Objetivo: Permitir que listas de inspeção sejam gerenciadas pelo sistema.

Requisitos: RU1, RU3, RU4, RU7.

Atores: Usuário logado.

Prioridade: Alta.

Trigger: O usuário logado seleciona a opção "fazer upload de listas" na sua página pessoal.

Fluxo Principal:

1. O sistema abre um pop-up para o usuário escolher arquivo.
2. O usuário escolhe um arquivo de sua máquina e clica no botão "upload" para fazer o upload.
3. O usuário clica no botão "visualizar listas de inspeção".
4. O sistema exibe uma tela de filtro com as seguintes informações.
 - Área de busca (text-input).
 - Filtros de data de criação [A1].
 - dois text-inputs de datas, ou dois date-pickers, para início e término dos períodos de filtro.
 - Filtros de data de modificação [A2].
 - dois text-inputs de datas, ou dois date-pickers, para início e término dos períodos de filtro.
 - Botão buscar.

Tabela com cada lista contendo:

- Nome da lista.
- Data de criação.
- Data da última modificação.
- Botão deletar.
- Botão editar.
- Botão visualizar histórico de inspeções.

5. O usuário preenche o campo "Área de busca" com o nome da lista adicionada e aciona o botão "Buscar".
6. O sistema exibe a mesma página com as listas filtradas pelo nome preenchido.
7. O usuário verifica que a lista adicionada consta na listagem.
8. O usuário aciona o botão "editar" para a lista recentemente adicionada.
9. O sistema exibe o editor padrão para a lista.
10. O usuário edita a lista.
11. O usuário clica no botão "salvar".
12. O sistema redireciona o usuário para a página de busca anterior.
13. O usuário aperta o botão "deletar".
14. O sistema requer uma confirmação do usuário para deletar a lista. [A2]
15. O usuário confirma a deleção e a lista é deletada do sistema.
16. O sistema redireciona o usuário para a página de busca anterior.

17. O usuário verifica que a lista deletada não aparece na página de busca.
18. O usuário clica no botão único em cada página que o redireciona para sua página inicial.
19. O sistema redireciona o usuário para sua página inicial.

Fluxos alternativos:

[A1]: O usuário preenche ambos os limites do campo "Filtros de datas de criação" com o dia atual e aciona o botão "Buscar":

1. O sistema exibe a mesma página com as listas filtradas pela data atual no campo "data de criação".
2. O fluxo segue segundo o passo 7 do fluxo principal.

[A2]: O usuário preenche ambos os limites do campo "Filtros de datas de modificação" com o dia atual e aciona o botão "Buscar":

1. O sistema exibe a mesma página com as listas filtradas pela data atual no campo "data da última modificação".
2. O fluxo segue segundo o passo 7 do fluxo principal.

Regras de Negócio: [RN1]: Todos os filtros de busca por data são opcionais.

UCU2 - Gerenciamento de documentos

UCU3 - Gerenciamento de inspeções

UCT1 - Gerenciamento de transeuntes

UCT2 - Gerenciamento de inspeções públicas

Timeline

- Terminar requisitos e timeline até dia 14/5.
- Fazer Gantt chart para timeline até dia 14/5.
- Dividir tarefas até dia 14/5.
- Terminar inspeção de requisitos até dia 14/5
- Terminar documento de projeto e arquitetura até dia 26/5.
- Terminar inspeção de documento de projeto de projeto e arquitetura 26/5
- Fazer sistema e página de login até dia 26/5 (UCLOGIN).
- Fazer uma de todos os artefatos até 26/5
- Terminar documentos de planejamento de testes até 4/6.
- Implementar C.R.U.D. e busca de listas de inspeção no backend até dia 4/6 (UCU1).
- Terminar listagem dos casos de teste até 23 de junho
- Fazer sistema de gerenciamento de documentos até 23/6 (UCU2).
- Fazer sistema de gerenciamento de inspeções até 23/6. (UCU3).
- Fazer sistema de gerenciamento de transeuntes até 23/6. (UCT1).
- Inspeccionar e testar a implementação até o momento, dia 23/6.
- Fazer slideshow de apresentação até dia 2/7.
- Terminar gerenciamento de inspeções públicas até dia 2/7 (UCT2).
- Finalizar pendências de outras fases até dia 2/7.
- Testes finais até dia 2/7.
- Entrega final.

Tarefas

Conclusão

Referências

[1]: Kalinowski, Marcos; Spínola, Rodrigo Oliveira, 2007, "Introdução à Inspeção de Software", Revista Engenharia de

Software, edição 1.

[2]: FAGAN, M.E., 1976, "Design and Code Inspection to Reduce Errors in Program Development", IBM Systems Journal, vol. 15, no. 3, pp. 182-211.