

# Requisitos do projeto Web – Princípios de Engenharia de Software

**Luiz Carlos R. Viana - 1810936**

**11 de Abril de 2020.**

luizcarlosrvn424@gmail.com

## 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, em 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 nós, do grupo 1 (G1), elegemos que o site a ser desenvolvido será uma aplicação voltada a inspetores profissionais, ou entusiastas, 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 intencionado por Fagan, e todos os seus atores.

## Requisitos

### Requisitos para Usuários

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

[RU2]: O usuário deve ser capaz de logar no sistema e gerenciar (C.R.U.D) suas listas de inspeção e documentos a serem inspecionados.

[RU3]: O usuário logado poderá editar suas listas de inspeção por um editor simples.

[RU4]: O usuário deve ser capaz de buscar listas de inspeção pelo nome e datas de criação e de última modificação.

[RU5]: Nenhum usuário deverá ser capaz de editar checklists de outros usuários.

### Requisitos para Transeuntes

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

[RT2]: Qualquer internauta poderá dar um like em uma lista de inspeção.

[RT3]: Nenhum internauta poderá editar o conteúdo de qualquer lista de inspeção sem estar logado.

[RT4]: Internautas devem poder criar contas de usuário.

[RT5]: Internautas devem poder se logar no sistema como usuários a partir da conta criada.

## Requisitos de Projeto

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

## Casos de uso

### UCCREATE - Criação de conta de Usuário

---

**Objetivo:** Permitir que internautas criem contas de usuário.

**Requisitos:** RT4.

**Atores:** Internauta anônimo.

**Prioridade:** Alta.

**Trigger:** O Internauta aciona o botão "criar conta" da página inicial.

#### Fluxo Principal:

1. O sistema apresenta um formulário de criação de conta (possivelmente envolvendo um redirect para outra página, ou não).
2. O usuário preenche: nome, login, senha, email, organização, e captcha. **[RN1]**
3. O sistema valida as informações e redireciona o usuário para a página inicial da conta recém criada. **[A1]**

#### Fluxos Alternativos:

**[A1]:** Se as informações preenchidas forem inválidas, o sistema gera uma mensagem de erro no formulário de criação de conta.

#### Regras de negócios:

**[RN1]:** Todos os campos, exceto organização, são obrigatórios e o email deve ser único.

### UCLOGIN - Login de Usuário

---

**Objetivo:** Permitir que internautas façam login como usuários.

**Requisitos:** RT5.

**Atores:** Internauta anônimo.

**Prioridade:** Alta.

**Trigger:** O Internauta, depois de criar uma conta, preenche o formulário de login na página inicial do site.

#### Fluxo Principal:

1. O internauta preenche seu login e senha definidos em UCCREATE.
2. O sistema valida o login e senha e redireciona o usuário para sua página inicial. **[A1]**

#### Fluxos Alternativos:

**[A1]:** Se o login e senha forem inválidos o sistema gera uma mensagem de erro na mesma página de login.

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

**Requisitos:** RU1, RU2, RU3, RU4.

**Atores:** Usuário logado.

**Prioridade:** Alta.

**Trigger:** O usuário logado seleciona a opção "criar de lista de inspeção" na sua página pessoal.

**Fluxo Principal:**

1. O sistema exibe um formulário para a criação de uma nova lista (possivelmente envolvendo um redirect para outra página).
2. O usuário preenche a área de texto com informações da nova lista.
3. O usuário aciona o botão "salvar".
4. O sistema registra a criação da nova lista e redireciona o usuário para sua página pessoal.
5. O usuário clica no botão "visualizar listas de inspeção".
6. O sistema exibe uma tela de filtro com as seguintes informações.
  - Área de busca (text-input).
  - Filtros de data de criação **[A1]** **[RN1]**.
    - dois text-inputs de datas, ou dois date-pickers, para início e término dos períodos de filtro.
  - Filtros de data de modificação **[A2]** **[RN1]**.
    - 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.
7. O usuário preenche o campo "Área de busca" com o nome da lista adicionada e aciona o botão "Buscar".
  8. O sistema exibe a mesma página com as listas filtradas pelo nome preenchido.
  9. O usuário verifica que a lista adicionada consta na listagem.
  10. O usuário aciona o botão "editar" para a lista recentemente adicionada.
  11. O sistema exibe o editor padrão para a lista.
  12. O usuário edita a lista.
  13. O usuário clica no botão "salvar".
  14. O sistema redireciona o usuário para a página de busca anterior.
  15. O usuário aperta o botão "deletar".
  16. O sistema requer uma confirmação do usuário para deletar a lista. **[A3]**

17. O usuário confirma a deleção e a lista é deletada do sistema.
18. O sistema redireciona o usuário para a página de busca anterior.
19. O usuário verifica que a lista deletada não aparece na página de busca.
20. O usuário clica no botão único em cada página que o redireciona para sua página pessoal.
21. O sistema redireciona o usuário para sua página pessoal.

**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.

**[A3]:** Se o usuário cancelar a deleção, ele permanece na página de filtro sem qualquer alteração.

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

**UCT1 - Gerenciamento de transeuntes**

---

**Objetivo:** Permitir que internautas acessem e dêem "like" em listas.

**Requisitos:** RT1, RT2.

**Atores:** Internauta anônimo.

**Prioridade:** Média.

**Trigger:** O Internauta aciona o botão "consulta geral" na página inicial do site.

**Fluxo Principal:**

1. O sistema exibe uma tela de filtro com as seguintes informações.
  - Área de busca (text-input).
  - Filtros de data de criação **[A1]** **[RN1]**.
    - dois text-inputs de datas, ou dois date-pickers, para início e término dos períodos de filtro.
  - Filtros de data de modificação **[A2]** **[RN1]**.
    - 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 curtir.
2. O usuário escolhe uma lista e aciona o seu botão "curtir".
  3. O sistema registra a curtida para aquela lista.

4. O usuário verifica que a lista foi curtida.

#### **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.

**[A3]:** Se o usuário cancelar a deleção, ele permanece na página de filtro sem qualquer alteração.

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

## **Conclusão**

Descrevemos sumariamente os requisitos do projeto. Seguindo o paradigma de desenvolvimento Ágil, este documento será versionado e adaptado futuramente segundo as necessidades do projeto, e servirá de base para o planejamento da sua arquitetura, implementação e testes.

## **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.