

# **Documentação do sistema**

## **1. Introdução**

Muitas vezes, as equipes de desenvolvimento utilizam ferramentas improvisadas para gerenciar requisitos de projetos, como editores de texto e planilhas. Apesar de ser possível editar esses documentos na nuvem em colaboração com a equipe, é difícil acompanhar as mudanças ao longo do projeto. Também é difícil de visualizar as classificações dos requisitos por diferentes critérios e visualizar as relações entre os requisitos. Também é difícil controlar o que cada usuário pode alterar no documento. Por fim, os usuários podem se distrair por questões relacionadas à formatação do documento.

Tendo em vista as dificuldades no gerenciamento de requisitos, foi idealizado o sistema WebReq. O objetivo é fornecer uma ferramenta colaborativa, poderosa e fácil de usar para o gerenciamento de requisitos, procurando superar as dificuldades encontradas em outras formas de gerenciamento de requisitos. O sistema terá a licença GNU Affero General Public License e poderá ser utilizado gratuitamente. O professor Victor Francisco Araya Santander está orientando o desenvolvimento do sistema e os alunos Marcelo Henrique Bittencourt e Sergio Souza Novak estão desenvolvendo o sistema.

O sistema será dividido em dois módulos principais: usuário e projeto. O módulo usuário será responsável por gerenciar o acesso ao sistema, enquanto o módulo projeto trará a funcionalidade principal do sistema. Os projetos serão divididos em informação, requisitos, colaboradores e configurações.

## **2. Especificação de requisitos**

### **2.1. Requisitos funcionais**

#### **[RF-01] Cadastrar usuário**

O sistema deverá permitir que qualquer usuário crie uma conta no sistema e informe seus dados, os quais deverão ser validados pelo sistema. Cada usuário deverá ter como atributos nome de usuário, senha, nome, e-mail, fuso horário, idioma, formação acadêmica e experiência profissional. A senha deverá ser armazenada de forma criptografada no banco de dados.

*Solicitante:* Victor.

*Prioridade:* Alta.

*Dependências:* Nenhuma.

#### **[RF-02] Consultar usuário**

O sistema deverá permitir a um usuário consultar todos os seus dados. Também deverá ser permitido ao usuário administrador do sistema consultar os dados de todos os usuários, com exceção da senha. Ao usuário cadastrado como gerente em um projeto, deverá ser permitido acessar as informações principais dos usuários participantes do projeto.

*Solicitante:* Victor.

*Prioridade:* Média.

*Dependências:* Fazer login.

#### **[RF-03] Atualizar usuário**

O sistema deverá permitir que o usuário atualize seus dados no sistema. Os novos dados deverão ser validados pelo sistema.

*Solicitante:* Victor.

*Prioridade:* Alta.

*Dependências:* Fazer login.

#### **[RF-04] Excluir usuário**

O sistema deverá permitir que o usuário exclua sua conta. Por questões de segurança, o sistema deverá solicitar que o usuário digite sua senha novamente. Ao administrador do sistema, deverá ser permitido excluir qualquer usuário.

*Solicitante:* Victor.

*Prioridade:* Média.

*Dependências:* Fazer login.

#### **[RF-05] Fazer login**

O sistema deverá permitir que o usuário se autentique no sistema. O usuário deverá informar nome de usuário e senha, que deverão ser validados pelo sistema. Se os dados estiverem corretos, o sistema deverá liberar ao usuário o acesso às funcionalidades do sistema conforme o seu perfil. O sistema deverá permitir usuários com os seguintes perfis:

- Usuário comum: Terá acesso somente aos seus dados e dados dos projetos dos quais participa.
- Usuário administrador: Terá acesso irrestrito a todas as informações do sistema.

No caso de usuário comum alocado como gerente de um projeto, o sistema deverá permitir que o mesmo possa adicionar, consultar, alterar e excluir usuários no projeto que o mesmo coordena. No caso de o usuário esquecer seu nome de usuário ou senha, o sistema deverá permitir que o usuário possa atualizar essas informações através do envio de mensagem para seu e-mail.

*Solicitante:* Victor.

*Prioridade:* Alta.

*Dependências:* Cadastrar usuário.

#### **[RF-06] Fazer logout**

O sistema deverá permitir que o usuário saia do sistema.

*Solicitante:* Victor.

*Prioridade:* Alta.

*Dependências:* Fazer login.

#### **[RF-07] Cadastrar projeto**

O sistema deverá permitir que o usuário crie um projeto e informe os dados do projeto, que deverão ser validados pelo sistema. Cada projeto deverá ter como atributos nome, data de início e data de fim. Cada projeto também deverá ter uma lista de colaboradores, com seus respectivos papéis no projeto. Os colaboradores deverão ser usuários já cadastrados no sistema. Quando for adicionar colaboradores ao projeto, o usuário criador do projeto deverá definir o papel exercido por cada colaborador. O sistema deverá permitir que o usuário criador do projeto possa criar papéis para cada projeto. O usuário criador do projeto deverá ser inicialmente alocado como gerente do projeto. Essa alocação poderá ser modificada posteriormente por ele.

*Solicitante:* Victor.

*Prioridade:* Alta.

*Dependências:* Fazer login.

#### **[RF-08] Consultar projeto**

O sistema deverá permitir que um usuário possa consultar todos os dados dos projetos dos quais participa.

*Solicitante:* Victor.

*Prioridade:* Alta.

*Dependências:* Cadastrar projeto.

#### **[RF-09] Atualizar projeto**

O sistema deverá permitir que o usuário com o perfil de gerente possa atualizar todos os dados de um projeto do qual ele participa. Os novos dados deverão ser validados pelo sistema.

*Solicitante:* Victor.

*Prioridade:* Alta.

*Dependências:* Cadastrar projeto.

#### **[RF-10] Excluir projeto**

O sistema deverá permitir que o usuário com perfil de gerente ou administrador possa excluir um projeto com todas as informações associadas. Por questões de segurança, o sistema deverá solicitar que o usuário digite sua senha novamente e digite o nome do projeto que ele deseja excluir.

*Solicitante:* Victor.

*Prioridade:* Média.

*Dependências:* Cadastrar projeto.

#### **[RF-11] Cadastrar requisito**

O sistema deverá permitir que um usuário alocado em um projeto com qualquer papel possa adicionar um novo requisito. O usuário deverá informar os dados do requisito, que deverão ser validados pelo sistema. Cada requisito deverá ter como atributos: identificador, descrição, data de criação, data da última modificação, solicitante, razão de existência do requisito, status, dependências em relação ao requisito, dependências do requisito em relação a outros requisitos, links para arquivos com mais detalhes, observações adicionais, categoria, versão, prioridade, complexidade e autor.

*Solicitante:* Victor.

*Prioridade:* Alta.

*Dependências:* Cadastrar projeto.

#### **[RF-12] Consultar requisito**

O sistema deverá permitir que um usuário colaborador de um projeto possa consultar todos os dados de cada requisito desse projeto. O usuário poderá consultar os requisitos por solicitante, por nível de prioridade, por palavra-chave contida na descrição do requisito, por status, por dependências e requisitos dependentes.

*Solicitante:* Victor.

*Prioridade:* Alta.

*Dependências:* Cadastrar requisito.

#### **[RF-13] Atualizar requisito**

O sistema deverá permitir que o usuário cadastrado como colaborador em um projeto possa atualizar qualquer dado de um requisito associado a esse projeto. O sistema deve manter um registro contendo data, hora e colaborador que efetuou a mudança no requisito. Esse registro poderá ser consultado por qualquer colaborador do projeto. Os novos dados devem ser validados pelo sistema. O atributo status do requisito só poderá ser modificado pelo gerente do projeto após um processo formal de validação.

*Solicitante:* Victor.

*Prioridade:* Alta.

*Dependências:* Cadastrar requisito.

#### **[RF-14] Excluir requisito**

O sistema deverá permitir que o usuário cadastrado como gerente de um projeto possa excluir um requisito do projeto. O usuário deve confirmar que realmente deseja fazer isso.

*Solicitante:* Victor.

*Prioridade:* Média.

*Dependências:* Cadastrar requisito.

#### **[RF-15] Adicionar colaborador**

O sistema deverá permitir que o usuário cadastrado como gerente de um projeto possa adicionar um colaborador ao projeto. O gerente também poderá definir as permissões do colaborador no projeto (somente leitura ou leitura e escrita) e poderá definir qual papel o colaborador desempenhará no projeto. O colaborador adicionado receberá um convite perguntando se ele deseja participar do projeto.

*Solicitante:* Victor.

*Prioridade:* Média.

*Dependências:* Cadastrar projeto.

#### **[RF-16] Consultar colaborador**

O sistema deverá permitir que qualquer colaborador de um projeto possa consultar os colaboradores do projeto, suas permissões e papel desempenhado.

*Solicitante:* Victor.

*Prioridade:* Média.

*Dependências:* Cadastrar colaborador.

#### **[RF-17] Redefinir permissões**

O sistema deverá gerenciar as permissões dos usuários cadastrados de acordo com seus papéis em projetos. Os papéis possíveis serão gerente e colaborador. As restrições serão sempre relativas ao projeto. Ao colaborador, será permitido: consultar projeto, cadastrar requisito, consultar requisito, atualizar requisito e consultar colaborador. Ao gerente de projeto, também será permitido: cadastrar projeto, atualizar projeto, excluir projeto, excluir requisito, adicionar colaborador, redefinir permissões e remover colaborador.

*Solicitante:* Victor.

*Prioridade:* Média.

*Dependências:* Cadastrar colaborador.

#### **[RF-18] Remover colaborador**

O sistema deverá permitir que o usuário gerente possa remover um colaborador de um projeto que ele gerencia. Além disso, o sistema também deverá permitir que um colaborador de um projeto saia do projeto. No entanto, sua saída deverá ser notificada ao gerente do projeto. Em ambos os casos, o usuário deverá confirmar que realmente deseja fazer isso.

*Solicitante:* Victor.

*Prioridade:* Média.

*Dependências:* Cadastrar colaborador.

## **2.2. Casos de uso**

## **Caso de uso 01: Cadastrar usuário**

### Informação característica

*Objetivo no contexto:* O sistema deverá permitir que o usuário crie uma conta no sistema.

*Escopo:* WebReq.

*Precondições:* Nenhuma.

*Condição final de sucesso:* Usuário cadastrado.

*Condição final de falha:* Usuário não cadastrado.

*Ator primário:* Usuário.

### Cenário principal de sucesso

1. O usuário informa seus dados.
2. Se os dados forem válidos, o usuário é cadastrado no sistema.

### Extensões

Não há.

### Informação relacionada

*Prioridade:* Alta.

*Desempenho alvo:* Em média, 2 minutos.

*Frequência:* Algumas vezes por mês.

*Caso de uso pai:* Nenhum.

*Casos de uso subordinados:* Nenhum.

*Atores secundários:* Nenhum.

## **Caso de uso 02: Consultar usuário**

### Informação característica

*Objetivo no contexto:* O sistema deverá listar os principais dados de cada usuário.

*Escopo:* WebReq.

*Precondições:* Usuário autenticado.

*Condição final de sucesso:* Usuários listados.

*Condição final de falha:* Não há.

*Ator primário:* Usuário.

#### Cenário principal de sucesso

1. O usuário acessa a página de usuários e visualiza os usuários, podendo selecionar algum deles para visualizar com mais detalhes.

#### Extensões

Não há.

#### Informação relacionada

*Prioridade:* Média.

*Desempenho alvo:* Em média, 15 segundos.

*Frequência:* Várias vezes por mês.

*Caso de uso pai:* Nenhum.

*Casos de uso subordinados:* Nenhum.

*Atores secundários:* Nenhum.

### **Caso de uso 03: Atualizar usuário**

#### Informação característica

*Objetivo no contexto:* O sistema deverá permitir que o usuário atualize seus dados no sistema.

*Escopo:* WebReq.

*Precondições:* Usuário autenticado.

*Condição final de sucesso:* Usuário atualizado.

*Condição final de falha:* Usuário não atualizado.



*Ator primário:* Usuário.

#### Cenário principal de sucesso

1. O usuário atualiza seus dados.
2. Se os dados forem válidos, o usuário é atualizado no sistema.

#### Extensões

Não há.

#### Informação relacionada

*Prioridade:* Alta.

*Desempenho alvo:* Em média, 1 minuto.

*Frequência:* Algumas vezes por mês.

*Caso de uso pai:* Nenhum.

*Casos de uso subordinados:* Nenhum.

*Atores secundários:* Nenhum.

### **Caso de uso 04: Excluir usuário**

#### Informação característica

*Objetivo no contexto:* O sistema deverá permitir que o usuário exclua sua conta.

*Escopo:* WebReq.

*Precondições:* Usuário autenticado.

*Condição final de sucesso:* Usuário excluído.

*Condição final de falha:* Usuário não excluído.

*Ator primário:* Usuário.

#### Cenário principal de sucesso

1. O usuário informa novamente sua senha.

2. Se a senha estiver correta, a conta de usuário é excluída.

#### Extensões

Não há.

#### Informação relacionada

*Prioridade:* Média.

*Desempenho alvo:* Em média, 30 segundos.

*Frequência:* Algumas vezes por ano.

*Caso de uso pai:* Nenhum.

*Casos de uso subordinados:* Nenhum.

*Atores secundários:* Nenhum.

### **Caso de uso 05: Fazer login**

#### Informação característica

*Objetivo no contexto:* O sistema deverá permitir que o usuário se autentique no sistema.

*Escopo:* WebReq.

*Precondições:* Usuário cadastrado.

*Condição final de sucesso:* Usuário autenticado.

*Condição final de falha:* Usuário não autenticado.

*Ator primário:* Usuário.

#### Cenário principal de sucesso

1. O usuário informa nome de usuário e senha.
2. Se os dados estiverem corretos, o usuário é autenticado e o acesso às demais funcionalidades do sistema é liberado para ele.

#### Extensões

Não há.

#### Informação relacionada

*Prioridade:* Alta.

*Desempenho alvo:* Em média, 30 segundos.

*Frequência:* Algumas vezes por dia.

*Caso de uso pai:* Nenhum.

*Casos de uso subordinados:* Nenhum.

*Atores secundários:* Nenhum.

### **Caso de uso 06: Fazer logout**

#### Informação característica

*Objetivo no contexto:* O sistema deverá permitir que o usuário saia do sistema.

*Escopo:* WebReq.

*Precondições:* Usuário autenticado.

*Condição final de sucesso:* Sessão encerrada.

*Condição final de falha:* Sessão não encerrada.

*Ator primário:* Usuário.

#### Cenário principal de sucesso

1. O sistema verifica se há dados não salvos.
2. Caso não haja dados não salvos ou o usuário escolha não salvar, o sistema encerra a sessão.

#### Extensões

Não há.

#### Informação relacionada

*Prioridade:* Alta.

*Desempenho alvo:* Em média, 15 segundos.

*Frequência:* Algumas vezes por dia.

*Caso de uso pai:* Nenhum.

*Casos de uso subordinados:* Nenhum.

*Atores secundários:* Nenhum.

## **Caso de uso 07: Cadastrar projeto**

### Informação característica

*Objetivo no contexto:* O sistema deverá permitir que o usuário crie um projeto.

*Escopo:* WebReq.

*Precondições:* Usuário autenticado.

*Condição final de sucesso:* Projeto cadastrado.

*Condição final de falha:* Projeto não cadastrado.

*Ator primário:* Usuário.

### Cenário principal de sucesso

1. O usuário informa os dados do projeto.
2. Se os dados forem válidos, o projeto é cadastrado.

### Extensões

Não há.

### Informação relacionada

*Prioridade:* Alta.

*Desempenho alvo:* Em média, 1 minuto.

*Frequência:* Algumas vezes por mês.

*Caso de uso pai:* Nenhum.

*Casos de uso subordinados:* Nenhum.

*Atores secundários:* Nenhum.

## **Caso de uso 08: Consultar projeto**

### Informação característica

*Objetivo no contexto:* O sistema deverá listar os principais dados de cada projeto.

*Escopo:* WebReq.

*Precondições:* Usuário autenticado.

*Condição final de sucesso:* Projetos listados.

*Condição final de falha:* Não há.

*Ator primário:* Usuário.

### Cenário principal de sucesso

1. O usuário acessa a página de projetos e visualiza os projetos existentes, podendo selecionar algum dos projetos para visualizar com mais detalhes.

### Extensões

Não há.

### Informação relacionada

*Prioridade:* Alta.

*Desempenho alvo:* Em média, 15 segundos.

*Frequência:* Várias vezes por dia.

*Caso de uso pai:* Nenhum.

*Casos de uso subordinados:* Nenhum.

*Atores secundários:* Nenhum.

## **Caso de uso 09: Atualizar projeto**

### Informação característica

*Objetivo no contexto:* O sistema deverá permitir que o usuário atualize os dados de um projeto.

*Escopo:* WebReq.

*Precondições:* O usuário deve ter permissão para isso.

*Condição final de sucesso:* Projeto atualizado.

*Condição final de falha:* Projeto não atualizado.

*Ator primário:* Usuário.

#### Cenário principal de sucesso

1. O usuário atualiza os dados do projeto.
2. Se os dados forem válidos, o projeto é atualizado.

#### Extensões

Não há.

#### Informação relacionada

*Prioridade:* Alta.

*Desempenho alvo:* Em média, 1 minuto.

*Frequência:* Algumas vezes por mês.

*Caso de uso pai:* Nenhum.

*Casos de uso subordinados:* Nenhum.

*Atores secundários:* Nenhum.

### **Caso de uso 10: Excluir projeto**

#### Informação característica

*Objetivo no contexto:* O sistema deverá permitir que o usuário exclua um projeto.

*Escopo:* WebReq.

*Precondições:* O usuário deve ter permissão para isso.

*Condição final de sucesso:* Projeto excluído.

*Condição final de falha:* Projeto não excluído.

*Ator primário:* Usuário.

#### Cenário principal de sucesso

1. O usuário preenche as informações de segurança.
2. Se os dados estiverem corretos, o projeto é excluído.

#### Extensões

Não há.

#### Informação relacionada

*Prioridade:* Média.

*Desempenho alvo:* Em média, 1 minuto.

*Frequência:* Algumas vezes por ano.

*Caso de uso pai:* Nenhum.

*Casos de uso subordinados:* Nenhum.

*Atores secundários:* Nenhum.

### **Caso de uso 11: Cadastrar requisito**

#### Informação característica

*Objetivo no contexto:* O sistema deverá permitir que o usuário adicione um requisito a um projeto.

*Escopo:* WebReq.

*Precondições:* O usuário deve ter permissão para isso.

*Condição final de sucesso:* Requisito cadastrado.

*Condição final de falha:* Requisito não cadastrado.

*Ator primário:* Usuário.

### Cenário principal de sucesso

1. O usuário informa os dados do requisito.
2. Se os dados forem válidos, o requisito é cadastrado.

### Extensões

Não há.

### Informação relacionada

*Prioridade:* Alta.

*Desempenho alvo:* Em média, 5 minutos.

*Frequência:* Algumas vezes por dia.

*Caso de uso pai:* Nenhum.

*Casos de uso subordinados:* Nenhum.

*Atores secundários:* Nenhum.

### **Caso de uso 12: Consultar requisito**

### Informação característica

*Objetivo no contexto:* O sistema deverá listar os principais dados de cada requisito do projeto.

*Escopo:* WebReq.

*Precondições:* O usuário deve ter permissão para isso.

*Condição final de sucesso:* Requisitos listados.

*Condição final de falha:* Não há.

*Ator primário:* Usuário.

### Cenário principal de sucesso

1. O usuário acessa a lista de requisitos de um projeto e visualiza os requisitos existentes, podendo selecionar algum dos requisitos para visualizar com mais detalhes.



### Extensões

Não há.

### Informação relacionada

*Prioridade:* Alta.

*Desempenho alvo:* Em média, 15 segundos.

*Frequência:* Várias vezes por dia.

*Caso de uso pai:* Nenhum.

*Casos de uso subordinados:* Nenhum.

*Atores secundários:* Nenhum.

## **Caso de uso 13: Atualizar requisito**

### Informação característica

*Objetivo no contexto:* O sistema deverá permitir que o usuário atualize os dados de um requisito de um projeto.

*Escopo:* WebReq.

*Precondições:* O usuário deve ter permissão para isso.

*Condição final de sucesso:* Requisito atualizado.

*Condição final de falha:* Requisito não atualizado.

*Ator primário:* Usuário.

### Cenário principal de sucesso

1. O usuário atualiza os dados do requisito.
2. Se os dados forem válidos, o requisito é atualizado.

### Extensões

Não há.

#### Informação relacionada

*Prioridade:* Alta.

*Desempenho alvo:* Em média, 2 minutos.

*Frequência:* Várias vezes por mês.

*Caso de uso pai:* Nenhum.

*Casos de uso subordinados:* Nenhum.

*Atores secundários:* Nenhum.

### **Caso de uso 14: Excluir requisito**

#### Informação característica

*Objetivo no contexto:* O sistema deverá permitir que o usuário exclua um requisito de um projeto.

*Escopo:* WebReq.

*Precondições:* O usuário deve ter permissão para isso.

*Condição final de sucesso:* Requisito excluído.

*Condição final de falha:* Requisito não excluído.

*Ator primário:* Usuário.

#### Cenário principal de sucesso

1. Se o usuário confirmar a exclusão, o requisito é excluído.

#### Extensões

Não há.

#### Informação relacionada

*Prioridade:* Média.

*Desempenho alvo:* Em média, 15 segundos.

*Frequência:* Algumas vezes por mês.

*Caso de uso pai:* Nenhum.

*Casos de uso subordinados:* Nenhum.

*Atores secundários:* Nenhum.

## **Caso de uso 15: Adicionar colaborador**

### Informação característica

*Objetivo no contexto:* O sistema deverá permitir que o usuário adicione um colaborador a um projeto.

*Escopo:* WebReq.

*Precondições:* O usuário deve ter permissão para isso.

*Condição final de sucesso:* Colaborador adicionado.

*Condição final de falha:* Colaborador não adicionado.

*Ator primário:* Usuário.

### Cenário principal de sucesso

1. O usuário informa o colaborador a ser adicionado.
2. Se o colaborador informado é um usuário válido, ele recebe um convite para participar do projeto.
3. Se o colaborador aceitar o convite, ele é adicionado ao projeto.

### Extensões

Não há.

### Informação relacionada

*Prioridade:* Média.

*Desempenho alvo:* Em média, 1 minuto.

*Frequência:* Várias vezes por mês.

*Caso de uso pai:* Nenhum.

*Casos de uso subordinados:* Nenhum.

*Atores secundários:* Nenhum.

## **Caso de uso 16: Consultar colaborador**

### Informação característica

*Objetivo no contexto:* O sistema deverá listar os colaboradores de um projeto e suas permissões.

*Escopo:* WebReq.

*Precondições:* O usuário deve ter permissão para isso.

*Condição final de sucesso:* Colaboradores listados.

*Condição final de falha:* Não há.

*Ator primário:* Usuário.

### Cenário principal de sucesso

1. O usuário acessa a lista de colaboradores de um projeto e visualiza os colaboradores, podendo selecionar algum deles para visualizar com mais detalhes.

### Extensões

Não há.

### Informação relacionada

*Prioridade:* Média.

*Desempenho alvo:* Em média, 15 segundos.

*Frequência:* Várias vezes por mês.

*Caso de uso pai:* Nenhum.

*Casos de uso subordinados:* Nenhum.

*Atores secundários:* Nenhum.

## **Caso de uso 17: Redefinir permissões**

### Informação característica

*Objetivo no contexto:* O sistema deverá permitir que o usuário redefina as permissões de um colaborador.

*Escopo:* WebReq.

*Precondições:* O usuário deve ter permissão para isso.

*Condição final de sucesso:* Permissões redefinidas.

*Condição final de falha:* Permissões não redefinidas.

*Ator primário:* Usuário.

### Cenário principal de sucesso

1. O usuário redefine as permissões do colaborador.

### Extensões

Não há.

### Informação relacionada

*Prioridade:* Média.

*Desempenho alvo:* Em média, 30 segundos.

*Frequência:* Algumas vezes por mês.

*Caso de uso pai:* Nenhum.

*Casos de uso subordinados:* Nenhum.

*Atores secundários:* Nenhum.

## **Caso de uso 18: Remover colaborador**

### Informação característica

*Objetivo no contexto:* O sistema deverá permitir que o usuário remova um colaborador de um projeto.

*Escopo:* WebReq.

*Precondições:* O usuário deve ter permissão para isso.

*Condição final de sucesso:* Colaborador removido.

*Condição final de falha:* Colaborador não removido.

*Ator primário:* Usuário.

#### Cenário principal de sucesso

1. Se o usuário confirmar a remoção, o colaborador é removido.

#### Extensões

Não há.

#### Informação relacionada

*Prioridade:* Média.

*Desempenho alvo:* Em média, 30 segundos.

*Frequência:* Algumas vezes por ano.

*Caso de uso pai:* Nenhum.

*Casos de uso subordinados:* Nenhum.

*Atores secundários:* Nenhum.

## **2.3. Diagramas**

Foram elaborados um diagrama de classes do sistema e um diagrama de sequência do requisito “Cadastrar usuário”. Os diagramas estão em anexo.

## **3. Planejamento**

### **3.1. *Product backlog***

<b>RF</b>	<b>Nome</b>	<b>Importância</b>	<b>Estimativa (horas)</b>	<b>Como demonstrar</b>
01	Cadastrar usuário	Alta	4	O usuário poderá se cadastrar no sistema, digitando seus dados, que deverão ser validados pelo sistema.
02	Consultar usuário	Média	4	O sistema deverá listar todos os usuários.
03	Atualizar usuário	Alta	4	O usuário poderá atualizar seus dados, que deverão ser validados novamente pelo sistema.
04	Excluir usuário	Média	4	O usuário poderá excluir sua conta após passar por uma verificação de segurança.
05	Fazer login	Alta	8	O usuário poderá se autenticar no sistema, digitando seu nome de usuário e senha.
06	Fazer logout	Alta	1	O usuário poderá sair do sistema.
07	Cadastrar projeto	Alta	4	O usuário poderá criar um projeto, digitando os dados do projeto, que deverão ser validados pelo sistema.
08	Consultar projeto	Alta	2	O sistema deverá listar todos os projetos, com os principais dados de cada um.
09	Atualizar projeto	Alta	2	O usuário poderá atualizar os dados de um projeto, que deverão ser validados novamente pelo sistema.
10	Excluir projeto	Média	2	O usuário poderá excluir um projeto após passar por uma verificação de segurança.

11	Cadastrar requisito	Alta	4	O usuário poderá criar um requisito, digitando os dados do requisito, que deverão ser validados pelo sistema.
12	Consultar requisito	Alta	2	O sistema deverá listar todos os requisitos de um projeto.
13	Atualizar requisito	Alta	2	O usuário poderá atualizar um requisito de um projeto.
14	Excluir requisito	Média	1	O usuário poderá excluir um requisito de um projeto.
15	Adicionar colaborador	Média	4	O usuário poderá adicionar um colaborador a um projeto.
16	Consultar colaborador	Média	2	O usuário poderá visualizar todos os colaboradores de um projeto.
17	Redefinir permissões	Média	8	O usuário poderá redefinir permissões para um colaborador de um projeto.
18	Remover colaborador	Média	2	O usuário poderá remover um colaborador de um projeto.

### 3.2. *Sprint backlog*

Foram feitas planilhas com o *sprint backlog* de todas as sprints ativas. Essas planilhas estão em anexo.

### 3.3. Reuniões diárias

#### **Sprint 1**

**14/09/2017**

**Local:** LCT – Bloco F – Unioeste Cascavel



**Hora:** 20h15

**Carga horária:** 2h

**O que eu fiz?**

*Marcelo:* Fizemos uma versão inicial do sistema.

*Sergio:* Idem.

**O que irei fazer?**

*Marcelo:* Iremos documentar os casos de uso e requisitos do sistema.

*Sergio:* Faremos a documentação do sistema (requisitos e casos de uso). Além disso, revisaremos outros documentos já preenchidos.

**Algum impedimento (técnico, gestão, pessoal, etc.) para eu fazer?**

*Marcelo:* Indisponibilidade do cliente para acompanhar o desenvolvimento da documentação.

*Sergio:* Idem.

**17/09/2017**

**Local:** Casa do Sergio

**Hora:** 9h15

**Carga horária:** 6h

**O que eu fiz?**

*Marcelo:* Iniciamos a documentação do sistema.

*Sergio:* Idem.

**O que irei fazer?**

*Marcelo:* Iremos implementar os itens da sprint.

*Sergio:* Iremos refatorar código já implementado e conferir se o que está implementado condiz com os requisitos. Depois disso, iremos implementar o que falta da sprint.

**Algum impedimento (técnico, gestão, pessoal, etc.) para eu fazer?**

*Marcelo:* Nenhum.

*Sergio:* Nenhum.

## **Sprint 2**

**19/09/2017**

**Local:** LCT – Bloco F – Unioeste Cascavel

**Hora:** 20h15

**Carga horária:** 2h

### **O que eu fiz?**

*Marcelo:* Implementamos o login, porém sem criptografia e sem armazenamento de sessão. Modularizamos o cabeçalho e o rodapé das páginas. Iniciamos a migração do Bootstrap 3 para a versão 4.

*Sergio:* Implementamos o login, e refatoramos o código a fim de encapsular a barra de navegação, o cabeçalho e o rodapé da página. Além disso, atualizamos o projeto diante da nova versão do *framework* Bootstrap.

### **O que irei fazer?**

*Marcelo:* Iremos implementar o cadastro de usuário.

*Sergio:* Idem.

### **Algum impedimento (técnico, gestão, pessoal, etc.) para eu fazer?**

*Marcelo:* Desconhecimento das funções criptográficas do PHP.

*Sergio:* Nenhum.

**24/09/2017**

**Local:** Casa do Sergio

**Hora:** 15h45

**Carga horária:** 4h

### **O que eu fiz?**

*Marcelo:* Fizemos o protótipo do cadastro e consulta de usuários.

*Sergio:* Idem.

**O que irei fazer?**

*Marcelo:* Iremos continuar a implementação do cadastro, consulta, atualização e exclusão de usuários.

*Sergio:* Idem.

**Algum impedimento (técnico, gestão, pessoal, etc.) para eu fazer?**

*Marcelo:* Nenhum.

*Sergio:* Nenhum.

**01/10/2017**

**Local:** Casa do Sergio

**Hora:** 16h15

**Carga horária:** 4h

**O que eu fiz?**

*Marcelo:* Começamos a configurar o cadastro de usuário no banco de dados.

*Sergio:* Idem.

**O que irei fazer?**

*Marcelo:* Iremos terminar de configurar o cadastro de usuário no banco de dados. Também iremos terminar de configurar o login.

*Sergio:* Idem.

**Algum impedimento (técnico, gestão, pessoal, etc.) para eu fazer?**

*Marcelo:* Nenhum.

*Sergio:* Nenhum.

**Sprint 3**

**04/10/2017**

**Local:** LCT – Bloco F – Unioeste Cascavel

**Hora:** 20h00

**Carga horária:** 2h

**O que eu fiz?**

*Marcelo:* Terminamos de implementar o cadastro de usuário, a listagem de usuários, o login e o logout.

*Sergio:* Idem.

**O que irei fazer?**

*Marcelo:* Iremos implementar a atualização e exclusão de usuário.

*Sergio:* Idem.

**Algum impedimento (técnico, gestão, pessoal, etc.) para eu fazer?**

*Marcelo:* Nenhum.

*Sergio:* Nenhum.

**12/10/2017**

**Local:** Casa da avó do Sergio

**Hora:** 16h00

**Carga horária:** 2h

**O que eu fiz?**

*Marcelo:* Iniciamos a implementação da atualização de usuário.

*Sergio:* Idem.

**O que irei fazer?**

*Marcelo:* Iremos continuar a implementação da atualização de usuário e também implementar a exclusão de usuário.

*Sergio:* Idem.

**Algum impedimento (técnico, gestão, pessoal, etc.) para eu fazer?**

*Marcelo:* Nenhum.

*Sergio:* Nenhum.

**14/10/2017**

**Local:** Casa do Sergio

**Hora:** 16h00

**Carga horária:** 2h

**O que eu fiz?**

*Marcelo:* Terminamos de implementar a atualização de usuário e a mudança de senha.

*Sergio:* Idem.

**O que irei fazer?**

*Marcelo:* Iremos implementar a exclusão de usuário.

*Sergio:* Idem.

**Algum impedimento (técnico, gestão, pessoal, etc.) para eu fazer?**

*Marcelo:* Nenhum.

*Sergio:* Nenhum.

**15/10/2017**

**Local:** Casa do Sergio

**Hora:** 14h45

**Carga horária:** 5h

**O que eu fiz?**

*Marcelo:* Dividimos a tela de configurações em abas, implementamos a exclusão de usuário e começamos a implementar a consulta de usuário.

*Sergio:* Idem.

**O que irei fazer?**

*Marcelo:* Iremos terminar de implementar a consulta de usuário. Também iremos definir os tipos de usuário e fazer as alterações necessárias no sistema. Além disso, iremos implementar a validação dos formulários e traduzir o sistema.

*Sergio:* Iremos terminar de implementar a consulta de usuário. Também iremos definir os tipos de usuário e fazer as alterações necessárias no sistema. Além disso, iremos implementar a validação dos formulários.

**Algum impedimento (técnico, gestão, pessoal, etc.) para eu fazer?**

*Marcelo:* Nenhum.

*Sergio:* Nenhum.

**Sprint 4**

**29/10/2017**

**Local:** Casa do Sergio

**Hora:** 10h00

**Carga horária:** 4h

**O que eu fiz?**

*Marcelo:* Terminamos de implementar a consulta de usuário e iniciamos a implementação da validação de formulários.

*Sergio:* Idem.

**O que irei fazer?**

*Marcelo:* Iremos estudar engenharia de requisitos. Também iremos implementar o CRUD de projetos.

*Sergio:* Idem.

**Algum impedimento (técnico, gestão, pessoal, etc.) para eu fazer?**

*Marcelo:* Nenhum.

*Sergio:* Nenhum.

**Sprint 6**

**25/11/2017**

**Local:** Casa do Marcelo

**Hora:** 13h45

**Carga horária:** 3h

**O que eu fiz?**

*Marcelo:* Começamos a estudar engenharia de requisitos.

*Sergio:* Idem.

**O que irei fazer?**

*Marcelo:* Iremos terminar de estudar engenharia de requisitos. Iremos passar a utilizar a abordagem TDD e o framework PHPUnit. Também iremos implementar o CRUD de projetos.

*Sergio:* Idem.

**Algum impedimento (técnico, gestão, pessoal, etc.) para eu fazer?**

*Marcelo:* Nenhum.

*Sergio:* Nenhum.

**Sprint 8**

**10/01/2018**

**Local:** Casa do Marcelo

**Hora:** 21h15

**Carga horária:** 1h

**O que eu fiz?**

*Marcelo:* Terminamos de estudar engenharia de requisitos. Começamos a testar o módulo de usuários.

*Sergio:* Idem.

### O que irei fazer?

*Marcelo:* Irei implementar o CRUD de projetos.

*Sergio:* Nada.

### Algum impedimento (técnico, gestão, pessoal, etc.) para eu fazer?

*Marcelo:* Nenhum.

*Sergio:* Estarei viajando.

## 4. Implementação

### 4.1. Linguagens

Foram utilizadas as seguintes linguagens:

- **CSS:** Linguagem de estilo utilizada no *framework* Bootstrap.
- **HTML:** Linguagem de marcação utilizada na estruturação das páginas do sistema.
- **JavaScript:** Linguagem de programação utilizada no lado do cliente.
- **PHP:** Linguagem de programação utilizada no lado do servidor.
- **SQL:** Linguagem de banco de dados utilizada nas requisições ao banco de dados.

### 4.2. Frameworks e bibliotecas

Foram utilizados os seguintes *frameworks* e bibliotecas:

- **Bootstrap:** *Framework* utilizado para a geração de componentes da interface.
- **jQuery:** Biblioteca utilizada para facilitar o acesso aos elementos da página nos códigos em JavaScript.
- **PHPUnit:** *Framework* utilizado para testes de unidade nos códigos em PHP.
- **Popper.js:** Biblioteca utilizada no *framework* Bootstrap.

### 4.3. Ferramentas



Foram utilizadas as seguintes ferramentas:

- **Git:** Gerenciador de versões.
- **GitHub:** Repositório.
- **Google Chrome:** Navegador de internet e interpretador de HTML, CSS e JavaScript.
- **LocMetrics:** Ferramenta geradora de métricas.
- **MySQL:** Programa para modelagem e acesso a bancos de dados.
- **NetBeans:** Ambiente de desenvolvimento integrado (IDE).
- **XAMPP:** Servidor web e interpretador de PHP.

## 5. Testes

### 5.1. Testes de unidade

Foram realizados testes de unidade com o *framework* PHPUnit nas funções de validação dos dados do usuário em PHP. O código foi aceito em todos os testes.

### 5.2. Métricas

Foi utilizada a ferramenta LocMetrics para gerar métricas simples sobre o projeto, como o número de arquivos e o número de linhas de código. As métricas geradas estão em anexo.

### 5.3. Plataforma alvo

O sistema foi desenvolvido para ser utilizado na web, como uma aplicação cliente-servidor. Além disso, a responsividade foi levada em conta, de forma que o sistema possa ser utilizado em dispositivos de tamanhos diferentes.

## 6. Arquitetura

Esta é a arquitetura do projeto:

```
doc
src
  main
    js
      controller
      model
    php
      controller
      model
      entities
      persistence
      view
        pages
        templates
        text
    sql
  third-party
    bootstrap
      css
      js
    jquery
    popper
test
  model
    entities
    user
```

A arquitetura do projeto foi baseada na arquitetura modelo-visão-controlador (MVC). A pasta principal do projeto está dividida em documentação, código-fonte e testes.

O código-fonte está dividido em principal e terceiros. A pasta principal de código-fonte está dividida por linguagem: JavaScript, PHP e SQL. A pasta de JavaScript está dividida em controlador e modelo. A pasta de PHP está dividida em controlador, modelo

e visão. O controlador está dividido em entidades e persistência (banco de dados). A visão está dividida em páginas, modelos e texto (traduções).

A pasta de códigos de terceiros contém o *framework* Bootstrap e as bibliotecas jQuery e Popper.js. A pasta do Bootstrap está dividida por linguagem: CSS e JavaScript.

A pasta de testes tem a divisão modelo. O modelo tem a divisão entidades. A pasta de entidades tem a divisão usuário.

## **7. Conclusão**

Apesar de vários requisitos terem sido implementados durante a disciplina, o sistema ainda não tem funcionalidades suficientes para que seja utilizado em projetos reais. Provavelmente, o sistema continuará sendo desenvolvido durante outras disciplinas para que possa, enfim, ser utilizado.

Entre os requisitos que ainda poderão ser implementados, estão o gerenciamento de versões dos requisitos, a rastreabilidade de requisitos e a adição de anexos aos requisitos. Além desses requisitos, outros poderão ser acrescentados ao sistema.