

WeddingWeb

Relatório de Software

(versão 1.0)

ÍNDICE

1.	INTRODUÇÃO	4
1.1	Domínio do Problema	4
1.2	Objetivo	4
2.	LEVANTAMENTO DE REQUISITOS	5
2.1	Requisitos funcionais	5
2.1.1	Usuários	5
[RF001]	Cadastrar Noivo(a)	5
[RF002]	Alterar Noivo(a)	5
[RF003]	Visualizar Noivo(a)	6
[RF004]	Excluir Noivo(a)	6
[RF005]	Cadastrar Profissional	7
[RF006]	Alterar Profissional	7
[RF007]	Visualizar Profissional	7
[RF008]	Excluir Profissional	8
[RF009]	Cadastrar Convidado	8
[RF010]	Alterar Convidado	8
[RF011]	Visualizar Convidado	9
[RF012]	Excluir Convidado	9
2.1.2	Login e Sessão	9
[RF013]	Efetuar <i>Login</i>	10
[RF014]	Efetuar <i>Logout</i>	10
2.1.3	Cerimônia e Organização	10
[RF015]	Cadastrar Casamento	10
[RF016]	Alterar Casamento	11
[RF017]	Visualizar Casamento	11
[RF018]	Excluir Casamento	12
[RF019]	Cadastrar Tarefa	12
[RF020]	Alterar Tarefa	12
[RF021]	Visualizar Tarefa	13
[RF022]	Excluir Tarefa	13
[RF023]	Cadastrar Serviço	14
[RF024]	Alterar Serviço	14
[RF025]	Visualizar Serviço	14
[RF026]	Excluir Serviço	15
[RF027]	Disponibilizar Serviço	15
[RF028]	Procurar Serviço	16
[RF029]	Procurar Profissional	16
[RF030]	Contratar Serviço	16
[RF031]	Gerenciar Tarefa	17
[RF032]	Gerenciar Serviços	17
2.1.4	Convites e Convidados	18
[RF033]	Cadastrar Convite	18
[RF034]	Alterar Convite	18
[RF035]	Visualizar Convite	18
[RF036]	Excluir Convite	19
[RF037]	Enviar Convite	19
[RF038]	Gerenciar Convidados	20
2.2	Requisitos não-funcionais	21
[NF001]	Usabilidade	21
[NF002]	Desempenho	21
[NF003]	Hardware e Software	21
2.3	Casos de uso	22
2.4	Diagrama de Classes	23
2.5	Modelo Relacional	23
3.	ARQUITETURA DO SISTEMA	25

3.1	Descrição da arquitetura do sistema	25
3.2	Tecnologias utilizadas	25
3.2.1	Servlets e JSP	25
	JSP é o acrônimo para Java Server Pages, uma linguagem criada pela SUN gratuita, JSP é uma linguagem de script com especificação aberta que tem como objetivo primário a geração de conteúdo dinâmico para páginas da Internet. Podemos ao invés de utilizar HTML para desenvolver páginas Web estáticas e sem funcionalidade, utilizar o JSP para criar dinamismo. É possível escrever HTML com códigos JSP embutidos. Como o HTML é uma linguagem estática, o JSP será o responsável por criar dinamismo. Por ser gratuita e possuir especificação aberta possui diversos servidores que suportam a linguagem, entre eles temos: Tomcat, GlassFish, JBoss, entre outros. O JSP necessita de servidor para funcionar por ser uma linguagem Server-side script, o usuário não consegue ver a codificação JSP, pois esta é convertida diretamente pelo servidor, sendo apresentado ao usuário apenas codificação HTML.	25
3.2.2	JPA e EclipseLink	26
3.2.3	JUnit	26
3.2.4	Spring Data JPA	26
3.3	Padrões de projetos	27
3.3.1	Repository	27
4.	TESTE DE SOFTWARE	28
4.1	Casos de testes	28
CT001	Realizar acesso ao sistema	28
CT002	Realizar encerramento de sessão do sistema	28
CT003	Realizar a criação de um casamento	29
CT004	Realizar a criação de uma tarefa	29
CT005	Realizar criação de usuário profissional	29
CT006	Realizar criação de um serviço	30
CT007	Realizar disponibilização de um serviço	30
CT008	Realizar contratação de um serviço	31
CT009	Realizar criação de um convite	31
CT010	Realizar o envio de um convite	32
CT011	Realizar o gerenciamento de convidados	32
CT012	Realizar o gerenciamento de tarefas	32
CT013	Realizar o gerenciamento de serviços	33
CT014	Realizar a procura de profissional	33
CT015	Realizar a procura de serviço	34
CT016	Responder um convite enviado	34
5.	CONCLUSÃO E TRABALHOS FUTUROS	36
6.	REFERÊNCIAS	37

1. INTRODUÇÃO

Apresentar uma visão geral breve sobre o sistema de software a que se refere este documento. Descrever os objetivos de desenvolvimento do sistema.

1.1 DOMÍNIO DO PROBLEMA

Planejar, organizar e executar uma cerimônia de casamento é uma atividade complexa, que, apesar de bastante frequente e comum, envolve diversas tarefas desde o planejamento, com certo tempo de antecedência, até a execução, no dia da cerimônia. Diante disso, o sistema de casamentos *Wedding Web* dispõe aos usuários (noivos e noivas prestes a se casarem) a possibilidade de gerir suas agendas de tarefas, coletiva e individualmente; gerenciar a contratação e pagamento de profissionais e serviços relacionados ao casamento, como *buffet* e fotografia, fornecendo um orçamento das contratações dos serviços; administrar a lista de convidados e automatizar o envio e gerência de convites.

O público-alvo desta ferramenta são pessoas físicas, mais especificamente, casais que estão planejando realizar uma cerimônia matrimonial

1.2 OBJETIVO

Dessa forma, o sistema web *WeddingWeb* se propõe auxiliar os usuários na gestão de todas as atividades pertinentes à cerimônia de casamento, de modo que visa atuar desde a concepção do casamento até sua efetiva realização, possibilitando aos seus usuários manterem o controle de todas as ações relacionadas à cerimônia de casamento e evitar problemas em quaisquer etapas da mesma.

2. LEVANTAMENTO DE REQUISITOS

Nesta seção são especificados os requisitos do sistema *WeddingWeb*, fornecendo aos desenvolvedores as informações necessárias para o projeto e implementação, assim como para a realização dos testes e homologação do sistema.

2.1 REQUISITOS FUNCIONAIS

Texto em que se vão identificando e apresentando os actores principais: perfis de utilizadores, aplicações directamente interessadas na funcionalidade oferecida pela aplicação e para os quais esta foi criada.

Descrição de actores secundários tb existentes por motivos de manutenção e supervisão do sistema.

2.1.1 Usuários

[RF001] Cadastrar Noivo(a)

Nome do caso de uso: Cadastrar Noivo(a)

Descrição do caso de uso: O sistema deve possibilitar ao usuário criar um perfil de usuário do tipo noivo(a).

Atores: Usuário não cadastrado

Regras de negócio:

Prioridade: ☒ **Essencial** ☐ **Importante** ☐ **Desejável**

Entradas e pré-condições: Usuário não ser cadastrado.

Saídas e pós-condição: Usuário cadastrado no sistema

[RF002] Alterar Noivo(a)

Nome do caso de uso: Alterar Noivo(a)

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo noivo(a) alterar os campos do seu perfil de usuário

Atores: Noivo(a)

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Entradas e pré-condições: Usuário cadastrado no sistema.

Saídas e pós-condição: Dados do perfil de usuário alterados.

[RF003] Visualizar Noivo(a)

Nome do caso de uso: Visualizar Noivo(a)

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo noivo(a) visualizar todos campos de seu perfil de usuário.

Atores: Noivo(a)

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Entradas e pré-condições: Usuário cadastrado.

Saídas e pós-condição: Exibição dos dados do usuário.

[RF004] Excluir Noivo(a)

Nome do caso de uso: Excluir Noivo(a)

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo noivo(a) excluir seu próprio perfil de usuário, de modo permanente.

Atores: Noivo(a)

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Entradas e pré-condições: Usuário cadastrado e logado no sistema.

Saídas e pós-condição: Usuário excluído e sessão encerrada.

[RF005] Cadastrar Profissional

Nome do caso de uso: Cadastrar Profissional

Descrição do caso de uso: O sistema deve possibilitar ao usuário criar um perfil de usuário do tipo profissional.

Atores: Usuário não cadastrado

Regras de negócio:

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições: Usuário não ser cadastrado.

Saídas e pós-condição: Usuário cadastrado no sistema

[RF006] Alterar Profissional

Nome do caso de uso: Alterar Profissional

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo profissional alterar os campos do seu perfil de usuário

Atores: Profissional

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições: Usuário cadastrado no sistema.

Saídas e pós-condição: Dados do perfil de usuário alterados.

[RF007] Visualizar Profissional

Nome do caso de uso: Visualizar Profissional

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo profissional visualizar todos campos de seu perfil de usuário.

Atores: Profissional

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições: Usuário cadastrado.

Saídas e pós-condição: Exibição dos dados do usuário.

[RF008] Excluir Profissional

Nome do caso de uso: Excluir Profissional

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo profissional excluir seu próprio perfil de usuário de modo permanente.

Atores: Profissional

Prioridade: ☐ Essencial ☒ **Importante** ☐ Desejável

Entradas e pré-condições: Usuário cadastrado e logado no sistema.

Saídas e pós-condição: Usuário excluído e sessão encerrada.

[RF009] Cadastrar Convidado

Nome do caso de uso: Cadastrar Convidado

Descrição do caso de uso: O sistema deve possibilitar ao usuário externo ao sistema, após aceitar o convite para uma cerimônia de casamento, criar um perfil de usuário do tipo convidado.

Atores: Usuário não cadastrado

Regras de negócio:

Prioridade: ☐ Essencial ☒ **Importante** ☐ Desejável

Entradas e pré-condições: Usuário não ser cadastrado.

Saídas e pós-condição: Usuário cadastrado no sistema

[RF010] Alterar Convidado

Nome do caso de uso: Alterar Convidado

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo convidado alterar os campos do seu perfil de usuário

Atores: Convidado

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Entradas e pré-condições: Usuário cadastrado no sistema.

Saídas e pós-condição: Dados do perfil de usuário alterados.

[RF011] Visualizar Convidado

Nome do caso de uso: Visualizar Convidado

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo convidado visualizar todos campos de seu perfil de usuário.

Atores: Convidado

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Entradas e pré-condições: Usuário cadastrado.

Saídas e pós-condição: Exibição dos dados do usuário.

[RF012] Excluir Convidado

Nome do caso de uso: Excluir Convidado

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo convidado excluir seu próprio perfil de usuário de modo permanente.

Atores: Convidado

Prioridade: ☐ Essencial ☒ **Importante** ☐ Desejável

Entradas e pré-condições: Usuário cadastrado e logado no sistema.

Saídas e pós-condição: Usuário excluído e sessão encerrada.

2.1.2 Login e Sessão

[RF013] Efetuar Login

Nome do caso de uso: Efetuar *Login*

Descrição do caso de uso: O sistema deve permitir ao usuário acessar sua conta (iniciar uma sessão), fazendo *login* com seu email e senha previamente cadastrados.

Atores: Usuário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Entradas e pré-condições: Estar cadastrado no sistema

Saídas e pós-condição: Sessão iniciada.

[RF014] Efetuar Logout

Nome do caso de uso: Efetuar *Logout*

Descrição do caso de uso: O sistema deve permitir ao usuário, uma vez que tenha efetuado *login*, encerrar sua sessão, fazendo *logout*.

Atores: Usuário

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Entradas e pré-condições: Sessão iniciada

Saídas e pós-condição: Sessão encerrada

2.1.3 Cerimônia e Organização

[RF015] Cadastrar Casamento

Nome do caso de uso: Cadastrar Casamento

Descrição do caso de uso: O sistema deve possibilitar ao usuário cadastrar um novo Casamento, que inclui, noivos, tarefas, serviços contratados e convidados

Atores: Noivo(a)

Regras de negócio: Cada casamento possui exatamente dois noivos.

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Entradas e pré-condições: Estar logado no sistema

Saídas e pós-condição: Novo casamento cadastrado

[RF016] Alterar Casamento

Nome do caso de uso: Alterar Casamento

Descrição do caso de uso: O sistema deve possibilitar ao usuário alterar um ou mesmo todos os campos do casamento.

Atores: Noivo(a)

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Entradas e pré-condições: Estar logado. Casamento cadastrado no sistema

Saídas e pós-condição: Casamento alterado.

[RF017] Visualizar Casamento

Nome do caso de uso: Visualizar Casamento

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo noivo(a) visualizar dados do casamento cadastrado ao qual ele está vinculado.

Atores: Noivo(a)

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Entradas e pré-condições: Existir casamento cadastrado no sistema

Saídas e pós-condição: Dados do casamento exibidos pelo sistema

[RF018] Excluir Casamento

Nome do caso de uso: Excluir Casamento

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo noivo(a) excluir definitivamente um casamento cadastrado.

Atores: Noivo(a)

Prioridade:

☐ Essencial

☒ Importante

☐ Desejável

Entradas e pré-condições: Existir casamento cadastrado no sistema

Saídas e pós-condição: Casamento excluído.

[RF019] Cadastrar Tarefa

Nome do caso de uso: Cadastrar Tarefa

Descrição do caso de uso: O sistema deve possibilitar ao usuário cadastrar uma nova tarefa, que contém: nome, descrição, data e status.

Atores: Noivo(a)

Prioridade:

☐ Essencial

☐ Importante

☐ Desejável

Entradas e pré-condições: Estar logado no sistema

Saídas e pós-condição: Nova tarefa cadastrada

[RF020] Alterar Tarefa

Nome do caso de uso: Alterar Tarefa

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo noivo(a) alterar um ou mesmo todos os campos da tarefa cadastrada

Atores: Noivo(a)

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Entradas e pré-condições: Estar logado. Tarefa cadastrada no sistema

Saídas e pós-condição: Tarefa alterada.

[RF021] Visualizar Tarefa

Nome do caso de uso: Visualizar Tarefa

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo noivo(a) visualizar dados da tarefa cadastrada.

Atores: Noivo(a)

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Entradas e pré-condições: Existir tarefa cadastrada no sistema

Saídas e pós-condição: Dados da tarefa exibidos pelo sistema

[RF022] Excluir Tarefa

Nome do caso de uso: Excluir Tarefa

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo noivo(a) excluir definitivamente uma tarefa cadastrada.

Atores: Noivo(a)

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Entradas e pré-condições: Existir tarefa cadastrada no sistema

Saídas e pós-condição: Tarefa excluída.

[RF023] Cadastrar Serviço

Nome do caso de uso: Cadastrar Serviço

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo profissional cadastrar um novo serviço, que contém: nome, descrição e valor.

Atores: Profissional

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições: Estar logado no sistema

Saídas e pós-condição: Novo serviço cadastrado

[RF024] Alterar Serviço

Nome do caso de uso: Alterar Serviço

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo profissional alterar um ou mesmo todos os campos do serviço cadastrado.

Atores: Profissional

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições: Estar logado. Serviço cadastrado no sistema

Saídas e pós-condição: Serviço alterado.

[RF025] Visualizar Serviço

Nome do caso de uso: Visualizar Serviço

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo profissional visualizar dados do serviço cadastrado.

Atores: Profissional

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Entradas e pré-condições: Existir serviço cadastrado no sistema

Saídas e pós-condição: Dados do serviço exibidos pelo sistema

[RF026] Excluir Serviço

Nome do caso de uso: Excluir Serviço

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo profissional excluir definitivamente um serviço cadastrado.

Atores: Profissional

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Entradas e pré-condições: Existir serviço cadastrado no sistema

Saídas e pós-condição: Serviço excluído.

[RF027] Disponibilizar Serviço

Nome do caso de uso: Disponibilizar Serviço

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo profissional disponibilizar um serviço para ser disponibilizado para contratação pelos usuários do tipo noivo(a).

Atores: Profissional

Prioridade: ☒ **Essencial** ☐ Importante ☐ Desejável

Entradas e pré-condições: Existir serviço cadastrado no sistema

Saídas e pós-condição: Serviço disponibilizado

[RF028] Procurar Serviço

Nome do caso de uso: Procurar Serviço

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo noivo(a) pesquisar por um serviço já disponibilizado e consultá-lo antes de contratá-lo ou não.

Atores: Noivo(a)

Prioridade: ☐ Essencial ☐ Importante ☒ Desejável

Entradas e pré-condições: Serviço disponibilizado no sistema

Saídas e pós-condição: Serviço consultado

[RF029] Procurar Profissional

Nome do caso de uso: Procurar Profissional

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo noivo(a) pesquisar por um profissional e consultar seus serviços disponibilizados.

Atores: Noivo(a)

Prioridade: ☐ Essencial ☐ Importante ☒ Desejável

Entradas e pré-condições: Profissional cadastrado no sistema

Saídas e pós-condição: Profissional consultado

[RF030] Contratar Serviço

Nome do caso de uso: Contratar Serviço

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo noivo(a) contratar um serviço

Atores: Noivo(a)

Prioridade: ☐ **Essencial** ☐ **Importante** ☐ **Desejável**

Entradas e pré-condições: Serviço disponibilizado no sistema

Saídas e pós-condição: Serviço contratado

[RF031] Gerenciar Tarefa

Nome do caso de uso: Gerenciar Tarefa

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo noivo(a) gerenciar uma tarefa, alterando seu campo status para concluído, cancelado ou pendente.

Atores: Noivo(a)

Prioridade: ☐ **Essencial** ☐ **Importante** ☐ **Desejável**

Entradas e pré-condições: Tarefa cadastrada no sistema

Saídas e pós-condição: Tarefa alterada

[RF032] Gerenciar Serviços

Nome do caso de uso: Gerenciar Serviços

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo noivo(a) gerenciar todos os serviços contratados, consultando o custo total dos serviços contratados.

Atores: Noivo(a)

Prioridade: ☐ **Essencial** ☐ **Importante** ☐ **Desejável**

Entradas e pré-condições: Serviços contratados pelo usuário do tipo noivo(a)

Saídas e pós-condição: Custo total dos serviços calculados

2.1.4 Convites e Convidados

[RF033] Cadastrar Convite

Nome do caso de uso: Cadastrar Convite

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo noivo(a) cadastrar um novo convite.

Atores: Noivo(a)

Regras de negócio: Cada casamento possui exatamente um convite.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições: Estar logado no sistema

Saídas e pós-condição: Novo convite cadastrado

[RF034] Alterar Convite

Nome do caso de uso: Alterar Convite

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo noivo(a) alterar um ou mesmo todos os campos do convite cadastrado.

Atores: Noivo(a)

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições: Estar logado. Convite cadastrado no sistema

Saídas e pós-condição: Convite alterado.

[RF035] Visualizar Convite

Nome do caso de uso: Visualizar Convite

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo noivo(a) visualizar dados do convite relacionado ao casamento ao qual ele está vinculado.

Atores: Noivo(a)

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

Entradas e pré-condições: Existir convite cadastrado no sistema

Saídas e pós-condição: Dados do convite exibidos pelo sistema

[RF036] Excluir Convite

Nome do caso de uso: Excluir Convite

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo noivo(a) excluir definitivamente um convite cadastrado.

Atores: Noivo(a)

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

Entradas e pré-condições: Existir convite cadastrado no sistema

Saídas e pós-condição: Convite excluído.

[RF037] Enviar Convite

Nome do caso de uso: Enviarr Convite

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo noivo(a) enviar um convite para usuários não cadastrados, inserindo os e-mails dos convidados. O Sistema deve enviar os convites para os e-mails relacionados no convite.

Atores: Noivo(a)

Prioridade: ☐ Essencial ☐ Importante ☒ Desejável

Entradas e pré-condições: Existir convite cadastrado no sistema

Saídas e pós-condição: Convite enviado.

[RF038] Gerenciar Convidados

Nome do caso de uso: Gerenciar Convidados

Descrição do caso de uso: O sistema deve possibilitar ao usuário do tipo noivo(a) gerenciar os convidados, visualizando os convidados que aceitaram o convite.

Atores: Noivo(a)

Prioridade:

☒ **Essencial**

☐ Importante

☐ Desejável

Entradas e pré-condições: Convite enviado para convidados.

Saídas e pós-condição: Convidados gerenciados.

2.2 REQUISITOS NÃO-FUNCIONAIS

[NF001] Usabilidade

A interface com o usuário é de vital importância para o sucesso do sistema. Principalmente por ser um sistema que não será utilizado diariamente, o usuário não possui tempo disponível para aprender como utilizar o sistema.

O sistema deverá possuir uma interface gráfica simples e sugestiva para o usuário primário sem se tornar cansativa aos usuários mais experientes. O usuário pouco experiente em computação deve ser capaz de utilizar qualquer funcionalidade do sistema em até 5 minutos.

Prioridade: ☒ **Essencial** ☐ **Importante** ☐ **Desejável**

[NF002] Desempenho

Embora não seja um requisito essencial ao sistema, deve ser considerada por corresponder a um fator de qualidade de software. Numa conexão estável, o sistema deve emitir uma resposta ao usuário de que está executando as ações (*feedback*), em no máximo 0,5 segundos.

Prioridade: ☐ **Essencial** ☒ **Importante** ☐ **Desejável**

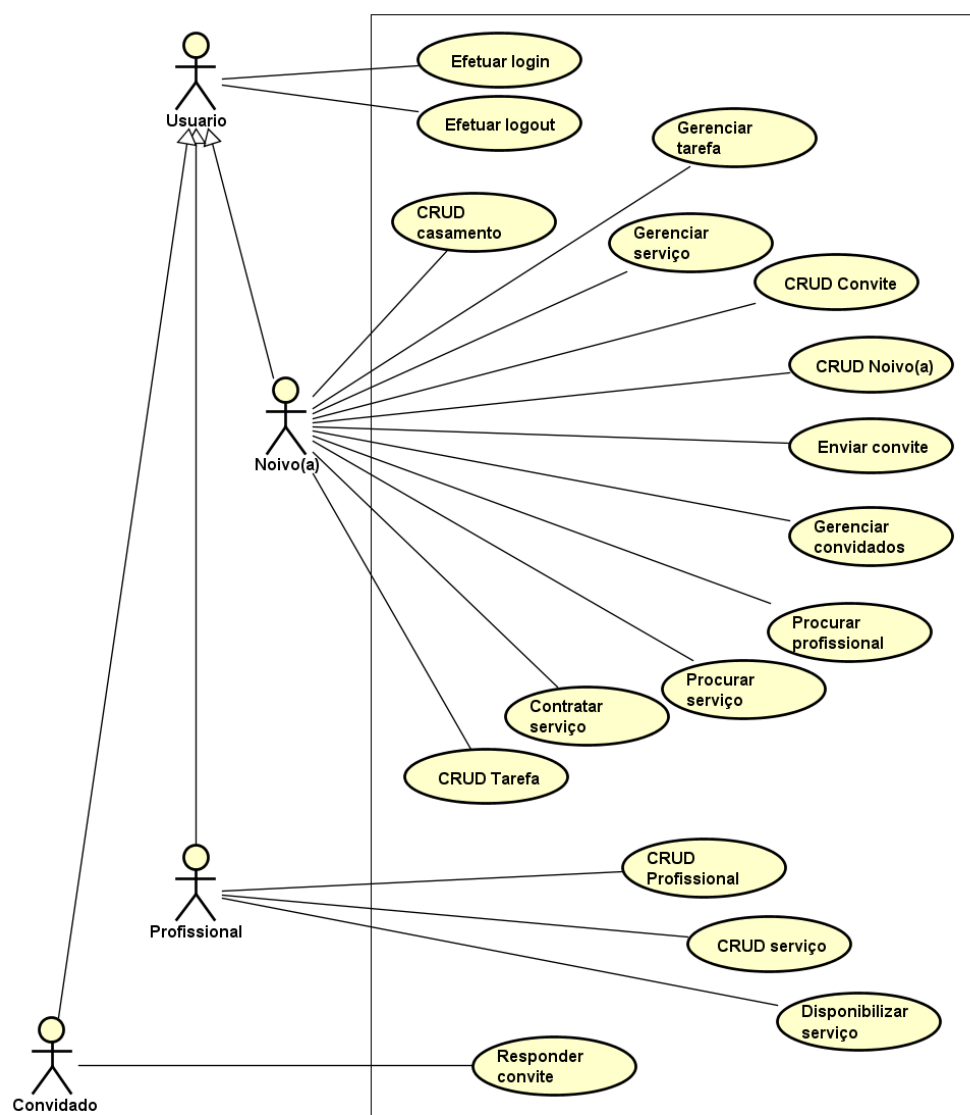
[NF003] Hardware e Software

O sistema deverá ser implementado utilizando, para o *front-end*, as linguagens HTML, CSS e *Java Script* e utilizando as tecnologias do *Java Enterprise Edition* (JEE), como *Servlets* e *JSP* (*JavaServer Pages*) para desenvolvimento do *back-end* e a implementação *Eclipse link* do *JPA* (*Java Persistence API*) para comunicação com o banco de dados.

Prioridade: ⊙ Essencial ⊙ Importante ⊙ Desejável

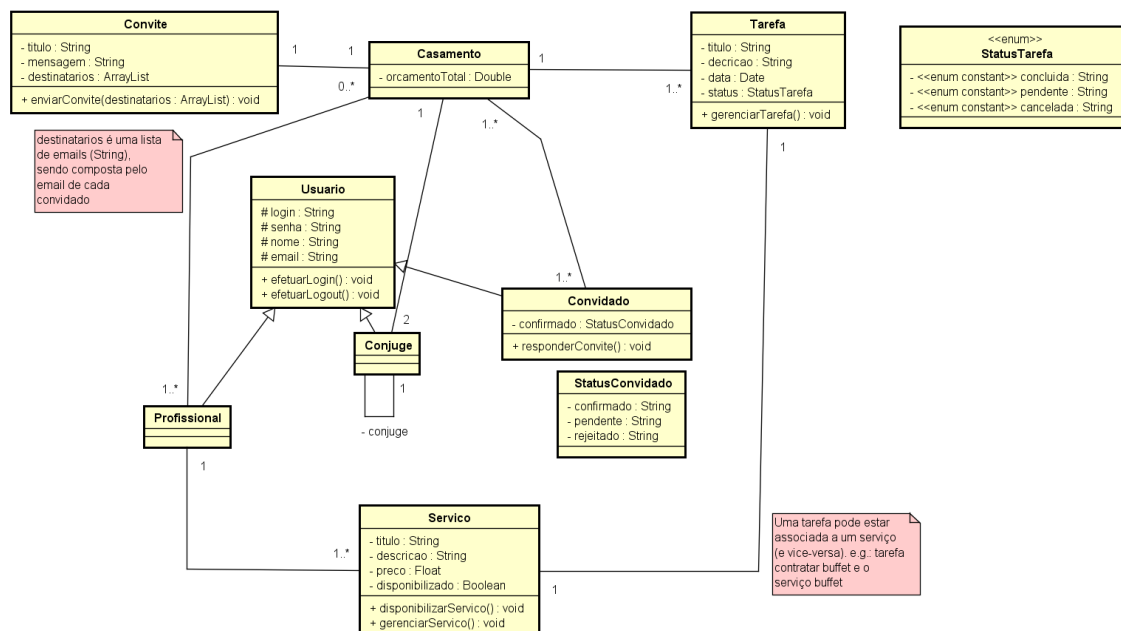
2.3 CASOS DE USO

O seguinte diagrama de casos de uso expõe os atores e casos de uso do sistema.



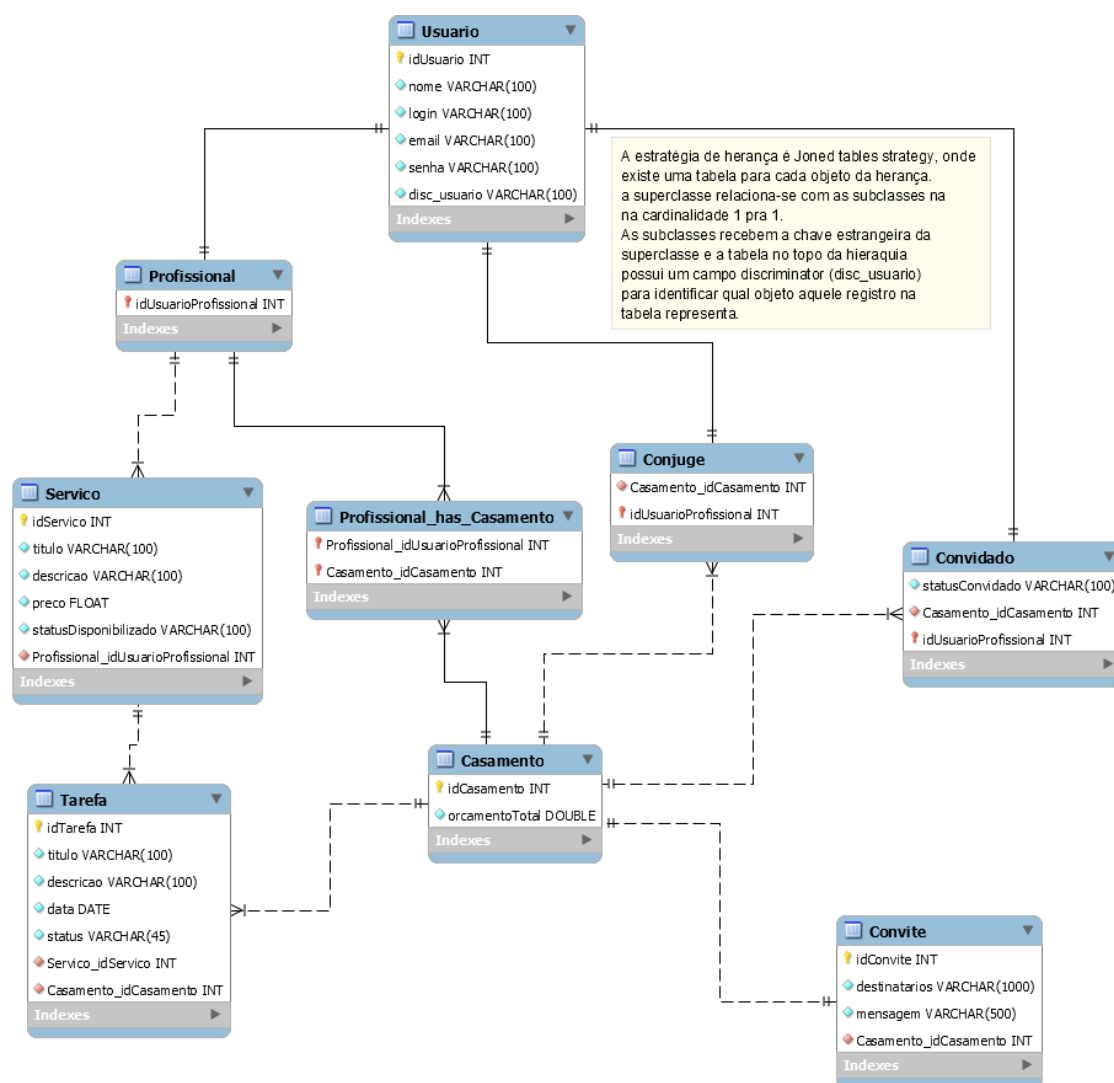
2.4 DIAGRAMA DE CLASSES

A estrutura de classes do sistema é representada pelo diagrama de classes abaixo:



2.5 MODELO RELACIONAL

A estrutura de tabelas do banco de dados do sistema pode ser representada pelo Modelo Relacional abaixo:



3. ARQUITETURA DO SISTEMA

3.1 DESCRIÇÃO DA ARQUITETURA DO SISTEMA

O sistema será implantado num servidor Apache Tomcat, versão 7, utilizando o SGBD MySQL, versão 5.7.

3.2 TECNOLOGIAS UTILIZADAS

Para o desenvolvimento do WeddingWeb, dentre todas as tecnologias utilizadas, destacam-se as principais:

3.2.1 Servlets e JSP

Servlets são a primeira forma que veremos de criar páginas dinâmicas com Java. Usaremos a própria linguagem Java para isso, criando uma classe que terá capacidade de gerar conteúdo HTML. O nome "servlet" vem da ideia de um pequeno servidor (*servidorzinho*, em inglês) cujo objetivo é receber chamadas HTTP, processá-las e devolver uma resposta ao cliente.

Uma primeira ideia da servlet seria que cada uma delas é responsável por uma página, sendo que ela lê dados da requisição do cliente e responde com outros dados (uma página HTML, uma imagem GIF etc). Como no Java tentamos sempre que possível trabalhar orientado a objetos, nada mais natural que uma servlet seja representada como um objeto a partir de uma classe Java.

Cada servlet é, portanto, um objeto Java que recebe tais requisições (request) e produz algo (response), como uma página HTML dinamicamente gerada.

JSP é o acrônimo para Java Server Pages, uma linguagem criada pela SUN gratuita, JSP é uma linguagem de script com especificação aberta que tem como objetivo primário a geração de conteúdo dinâmico para páginas da Internet. Podemos ao invés de utilizar HTML para desenvolver páginas Web estáticas e sem funcionalidade, utilizar o JSP para criar dinamismo. É possível escrever HTML com códigos JSP embutidos. Como o HTML é uma linguagem estática, o JSP será o responsável por criar dinamismo. Por ser gratuita e possuir especificação aberta possui diversos servidores que suportam a linguagem, entre eles temos: Tomcat, GlassFish, JBoss, entre outros. O JSP necessita de servidor para funcionar por ser uma linguagem Server-side script, o usuário não consegue ver a codificação JSP, pois esta é convertida diretamente pelo servidor, sendo apresentado ao usuário apenas codificação HTML.

3.2.2 JPA e EclipseLink

O processo de mapear objetos Java para tabelas em Banco de Dados e vice-versa é chamado de "Object-Relational Mapping" (ORM) ou Mapeamento Objeto-Relacional. A JPA é uma das possibilidades para o Mapeamento Objeto-Relacional.

JPA é uma especificação e atualmente várias implementações estão disponíveis, entre elas Hibernate, EclipseLink e Apache OpenJPA. No entanto, a implementação de referência da JPA é o EclipseLink, ou seja, todos fornecedores podem se basear no EclipseLink para criar suas próprias implementações. EclipseLink também garante que tudo que está especificado na JPA está implementado no EclipseLink. JPA também permite ao desenvolvedor mapear, armazenar, atualizar e recuperar informações de uma base de dados relacional para objetos Java e o contrário também é permitido.

3.2.3 JUnit

JUnit é um Framework open-source utilizado para facilitar o desenvolvimento de códigos em Java verificando se os resultados gerados pelos métodos são os esperados. Caso não sejam, o JUnit exibe os possíveis erros que estão ocorrendo nos métodos. Essa verificação é chamada de teste unitário ou teste de unidade.

Atualmente, buscando cada vez mais melhorias nos softwares, os desenvolvedores fazem uma bateria de testes nos seus códigos. Um desses testes é o teste de unidade que testa a menor parte do código garantindo uma maior qualidade do produto no processo de desenvolvimento. No caso da linguagem Java esse teste é feito através do JUnit em cada método separadamente.

3.2.4 Spring Data JPA

O Spring Data JPA é um framework que nasceu para facilitar a criação dos repositórios (ou DAOS). Ele faz isso nos liberando de ter que implementar as interfaces referentes aos nossos repositórios e também já deixando pré-implementado algumas funcionalidades como, por exemplo, de ordenação das consultas e de paginação de registros.

O Spring Data JPA é, na verdade, um projeto dentro de um outro maior que é o Spring Data. O Spring Data tem por objetivo facilitar nosso trabalho com persistência de dados de uma forma geral. E além do Spring Data JPA, ele possui vários outros projetos.

3.3 PADRÕES DE PROJETOS

O sistema fez uso dos seguintes padrões de projetos orientados a objetos:

3.3.1 Repository

Um repositório está vinculado à regra de negócio da aplicação e está associado ao agregado dos seus objetos de negócio e retorna objetos de domínio que representam esses dados.

Já o DAO (*Data Access Object*) a princípio tem o seu escopo limitado na captura e persistência dos dados de um objeto DTO (*Data Transfer Object*) que representa um registro da base de dados, conseqüentemente ele transmite apenas o mundo físico relacional da base de dados e não representa o minimundo real do negócio da sua aplicação.

Por fim, o repository é regra de negócio, engloba tudo sobre essa regra, retornando objetos de domínio do negócio

3.3.2 Abstract Factory

Abstract Factory é um padrão de projeto de software (*também conhecido como design pattern* em inglês). Este padrão permite a criação de famílias de objetos relacionados ou dependentes por meio de uma única interface e sem que a classe concreta seja especificada. Uma fábrica é a localização de uma classe concreta no código em que objetos são construídos. O objetivo em empregar o padrão é isolar a criação de objetos de seu uso e criar famílias de objetos relacionados sem ter que depender de suas classes concretas. Isto permite novos tipos derivados de ser introduzidas sem qualquer alteração ao código que usa a classe base. O uso deste padrão torna possível trocar implementações concretas sem alterar o código que estas usam, mesmo em tempo de execução.

3.3.3 Singleton

O Padrão Singleton tem como definição garantir que uma classe tenha apenas uma instância de si mesma e que forneça um ponto global de acesso a ela. Ou seja, uma classe gerencia a própria instância dela além de evitar que qualquer outra classe crie uma instância dela. Para criar a instancia tem-se que passar pela classe obrigatoriamente, nenhuma outra classe pode instanciar ela. O Padrão Singleton também oferece um ponto global de acesso a sua instância. A própria classe sempre vai oferecer a própria instância dela e caso não tenha ainda uma instância, então ela mesma cria e retorna essa nova instância criada.

4. TESTE DE SOFTWARE

Descrição textual e em diagrama da arquitectura lógica e física do sistema. Justificar as opções tomadas.

Figura 1. Arquitectura Lógica (diagrama de classes com os subsistemas todos do sistema)

Figura 2. Arquitectura Física (diagrama de componentes e de distribuição do sistema)

4.1 CASOS DE TESTES

Para a definição dos Casos de Teste foram utilizados os módulos desenvolvidos e a modelagem do banco de dados do sistema WeddingWeb.

ID DO CASO DE TESTE	CT001 – REALIZAR ACESSO AO SISTEMA
DESCRIÇÃO	Verificar se o usuário consegue efetuar o login no sistema.
ITEM A SER TESTADO	Caso de uso Efetuar login
CARACTERÍSTICAS A SEREM TESTADAS	Funcionalidade
ENTRADAS	<ul style="list-style-type: none"> • Usuário: noivo1 • Senha: 123456789
RESULTADOS E COMPORTAMENTOS ESPERADOS	<ul style="list-style-type: none"> • Mensagem na tela: “Login efetuado com sucesso” • Redirecionar para a tela principal (consultar tarefas)

ID DO CASO DE TESTE	CT002 – REALIZAR ENCERRAMENTO DE SESSÃO DO SISTEMA
DESCRIÇÃO	Verificar se o usuário consegue efetuar o logout no sistema.
ITEM A SER TESTADO	Caso de uso Efetuar logout
CARACTERÍSTICAS A SEREM TESTADAS	Funcionalidade
ENTRADAS	<ul style="list-style-type: none"> • Clicar na opção logout
RESULTADOS E COMPORTAMENTOS	<ul style="list-style-type: none"> • Redirecionar para a tela inicial (tela de login)

ESPERADOS	
-----------	--

ID DO CASO DE TESTE	CT003 – REALIZAR A CRIAÇÃO DE UM CASAMENTO
DESCRIÇÃO	Testar se um casamento foi criado no sistema.
ITEM A SER TESTADO	Caso de uso Criar casamento
CARACTERÍSTICAS A SEREM TESTADAS	Funcionalidade
ENTRADAS	<ul style="list-style-type: none"> Clicar na cadastrar-se Preencher campos do casamento, noivo e noiva Clicar na opção cadastrar
RESULTADOS E COMPORTAMENTOS ESPERADOS	<ul style="list-style-type: none"> Mensagem na tela: Casamento criado com sucesso

ID DO CASO DE TESTE	CT004 – REALIZAR A CRIAÇÃO DE UMA TAREFA
DESCRIÇÃO	Verificar se uma tarefa foi criada no sistema.
ITEM A SER TESTADO	Caso de uso Criar tarefa
CARACTERÍSTICAS A SEREM TESTADAS	Funcionalidade
ENTRADAS	<ul style="list-style-type: none"> Tela principal: Clicar na opção criar tarefa Preencher campos da tarefa Clicar na opção salvar
RESULTADOS E COMPORTAMENTOS ESPERADOS	<ul style="list-style-type: none"> Mensagem na tela: Tarefa criada com sucesso

ID DO CASO DE TESTE	CT005 – REALIZAR CRIAÇÃO DE USUÁRIO PROFISSIONAL
DESCRIÇÃO	Verificar se um usuário do tipo profissional é criado no sistema
ITEM A SER TESTADO	Caso de uso Criar profissional
CARACTERÍSTICAS A	Funcionalidade

SEREM TESTADAS	
ENTRADAS	<ul style="list-style-type: none"> • Clicar na opção cadastrar-se como profissional • Preencher campos do usuário
RESULTADOS E COMPORTAMENTOS ESPERADOS	<ul style="list-style-type: none"> • Mensagem na tela: Profissional criado com sucesso • Redirecionar para a tela inicial (tela de login)

ID DO CASO DE TESTE	CT006 – REALIZAR CRIAÇÃO DE UM SERVIÇO
DESCRIÇÃO	Verificar se um serviço foi criado no sistema
ITEM A SER TESTADO	Caso de uso Criar serviço
CARACTERÍSTICAS A SEREM TESTADAS	Funcionalidade
ENTRADAS	<ul style="list-style-type: none"> • Estar na tela principal • Clicar na opção serviços • Clicar na opção cadastrar serviço • Preencher campos do serviço • Clicar em salvar
RESULTADOS E COMPORTAMENTOS ESPERADOS	<ul style="list-style-type: none"> • Mensagem na tela: Serviço criado com sucesso • Redirecionar para a tela de serviços

ID DO CASO DE TESTE	CT007 – REALIZAR DISPONIBILIZAÇÃO DE UM SERVIÇO
DESCRIÇÃO	Verificar se um serviço cadastrado está disponibilizado
ITEM A SER TESTADO	Caso de uso Disponibilizar serviço
CARACTERÍSTICAS A SEREM TESTADAS	Funcionalidade
ENTRADAS	<ul style="list-style-type: none"> • Estar na tela de serviços • Clicar em um serviço • Selecionar a opção disponibilizar serviço
RESULTADOS E COMPORTAMENTOS	

ESPERADOS	<ul style="list-style-type: none"> • Mostrar na tela: Serviço disponibilizado com sucesso • Redirecionar para a tela de serviços
-----------	--

ID DO CASO DE TESTE	CT008 – REALIZAR CONTRATAÇÃO DE UM SERVIÇO
DESCRIÇÃO	Verificar se um serviço é contratado no sistema
ITEM A SER TESTADO	Caso de uso Contratar serviço
CARACTERÍSTICAS A SEREM TESTADAS	Funcionalidade
ENTRADAS	<ul style="list-style-type: none"> • Estar na tela de tarefas • Selecionar uma tarefa • Clicar na opção contratar serviço • Procurar serviço • Clicar na opção contratar serviço
RESULTADOS E COMPORTAMENTOS ESPERADOS	<ul style="list-style-type: none"> • Mostrar na tela: Serviço contratado com sucesso • Redirecionar para a tela de consulta de tarefas

ID DO CASO DE TESTE	CT009 – REALIZAR CRIAÇÃO DE UM CONVITE
DESCRIÇÃO	Verificar se um convite foi criado no sistema
ITEM A SER TESTADO	Caso de uso Criar convite
CARACTERÍSTICAS A SEREM TESTADAS	Funcionalidade
ENTRADAS	<ul style="list-style-type: none"> • Estar na tela principal • Clicar na opção convites • Clicar na opção criar convite • Preencher campos do convite • Clicar na opção salvar
RESULTADOS E COMPORTAMENTOS ESPERADOS	<ul style="list-style-type: none"> • Mensagem na tela: Convite criado com sucesso • Redirecionar para a tela de convite

ID DO CASO DE TESTE	CT010 – REALIZAR O ENVIO DE UM CONVITE
DESCRIÇÃO	Verificar se um convite foi enviado para o usuário
ITEM A SER TESTADO	Caso de uso Enviar convite
CARACTERÍSTICAS A SEREM TESTADAS	Funcionalidade
ENTRADAS	<ul style="list-style-type: none"> • Estar na tela principal • Clicar na opção convites • Selecionar o convite • Clicar na opção Enviar convite
RESULTADOS E COMPORTAMENTOS ESPERADOS	<ul style="list-style-type: none"> • Mensagem na tela: Convite enviado com sucesso • Redirecionar para a tela de convite

ID DO CASO DE TESTE	CT011 – REALIZAR O GERENCIAMENTO DE CONVIDADOS
DESCRIÇÃO	Verificar se os convidados podem ser gerenciados
ITEM A SER TESTADO	Caso de uso Gerenciar convidados
CARACTERÍSTICAS A SEREM TESTADAS	Funcionalidade
ENTRADAS	<ul style="list-style-type: none"> • Estar na tela principal • Clicar na opção convidados
RESULTADOS E COMPORTAMENTOS ESPERADOS	<ul style="list-style-type: none"> • Sistema exibe os convidados com os status correspondentes (confirmado, pendente, rejeitado) de acordo com o a aceitação do convite já enviado

ID DO CASO DE TESTE	CT012 – REALIZAR O GERENCIAMENTO DE TAREFAS
DESCRIÇÃO	Verificar se as tarefas podem ser gerenciados
ITEM A SER TESTADO	Caso de uso Gerenciar tarefas

CARACTERÍSTICAS A SEREM TESTADAS	Funcionalidade
ENTRADAS	<ul style="list-style-type: none"> Estar na tela principal Selecionar uma tarefa Clicar em gerenciar tarefa Alterar status da tarefa
RESULTADOS E COMPORTAMENTOS ESPERADOS	<ul style="list-style-type: none"> Sistema altera o status da tarefa selecionada

ID DO CASO DE TESTE	CT013 – REALIZAR O GERENCIAMENTO DE SERVIÇOS
DESCRIÇÃO	Verificar se os serviços podem ser gerenciados
ITEM A SER TESTADO	Caso de uso Gerenciar serviços
CARACTERÍSTICAS A SEREM TESTADAS	Funcionalidade
ENTRADAS	<ul style="list-style-type: none"> Estar na tela principal Clicar na opção serviços Selecionar um serviço Disponibilizar serviço, alterar valores Clicar na opção salvar
RESULTADOS E COMPORTAMENTOS ESPERADOS	<ul style="list-style-type: none"> Sistema altera serviço selecionado Sistema redireciona para a tela de serviços

ID DO CASO DE TESTE	CT014 – REALIZAR A PROCURA DE PROFISSIONAL
DESCRIÇÃO	Verificar se os profissionais podem procurados e encontrados
ITEM A SER TESTADO	Caso de uso Procurar profissional
CARACTERÍSTICAS A SEREM TESTADAS	Funcionalidade
ENTRADAS	

	<ul style="list-style-type: none"> Estar na tela principal Clicar na opção serviços Selecionar a opção procurar profissional Digitar nome do profissional e clicar em pesquisar
RESULTADOS E COMPORTAMENTOS ESPERADOS	<ul style="list-style-type: none"> Sistema busca profissional pelo nome Se encontrar, exibe na tela: "Profissional encontrado" e mostra os serviços do profissional Caso não encontre, exibe na tela: "Não foi possível encontrar o profissional"

ID DO CASO DE TESTE	CT015 – REALIZAR A PROCURA DE SERVIÇO
DESCRIÇÃO	Verificar se os serviços podem procurados e encontrados
ITEM A SER TESTADO	Caso de uso Procurar serviços
CARACTERÍSTICAS A SEREM TESTADAS	Funcionalidade
ENTRADAS	<ul style="list-style-type: none"> Estar na tela principal Clicar na opção serviços Selecionar a opção procurar serviço Digitar nome do serviço e clicar em pesquisar
RESULTADOS E COMPORTAMENTOS ESPERADOS	<ul style="list-style-type: none"> Sistema busca serviço pelo título Se encontrar, exibe na tela: "Serviço encontrado com sucesso" e mostra o serviço Caso não encontre, exibe na tela: "Não foi possível encontrar o serviço pesquisado"

ID DO CASO DE TESTE	CT016 – RESPONDER UM CONVITE ENVIADO
DESCRIÇÃO	Verificar se os convites podem respondidos

ITEM A SER TESTADO	Caso de uso Procurar serviços
CARACTERÍSTICAS A SEREM TESTADAS	Funcionalidade
ENTRADAS	<ul style="list-style-type: none"> • Ser usuário convidado • Clicar no link recebido por email • Clicar na resposta desejada • Clicar em salvar
RESULTADOS E COMPORTAMENTOS ESPERADOS	<ul style="list-style-type: none"> • Sistema recebe resposta do convite • Caso resposta seja confirmado: • Cadastrar-se como convidado • Caso resposta seja rejeitado: • Redirecionar para a tela inicial do sistema

5. CONCLUSÃO E TRABALHOS FUTUROS

O desenvolvimento do sistema WeddingWeb abre as portas para a expansão do mesmo, dado o surgimento de possibilidades e campos que esse sistema pode abranger na gestão de casamentos, suas tarefas, serviços, finanças, convidados, dentre outras coisas.

Se considerarmos as plataformas móveis, pode-se alcançar uma maior adesão ao uso, tratando do desenvolvimento de um sistema móvel e um sistema web, com todas as tecnologias disponibilizadas (muitas até *open-source*) atualmente.

6. REFERÊNCIAS

Caleum. **Servlets – Java para Desenvolvimento Web**. Disponível em <<https://www.caelum.com.br/apostila-java-web/servlets/>> Acesso em: 11 de dez. 2017.

DEVMEDIA. **Introdução ao Java Server Pages – JSP**. Disponível em <<https://www.devmedia.com.br/introducao-ao-java-server-pages-jsp/25602/>>. Acesso em: 11 de dez. 2017.

Wikidot. **Junit – Testes de Unidade**. Disponível em <<http://junit.wikidot.com/>>. Acesso em: 11 de dez. 2017.

Algaworks. **O que é Spring Data JPA**. Disponível em <<http://blog.algaworks.com/spring-data-jpa/>> Acesso em: 11 de dez. 2017.