##### Requisitos do projeto Organização de Eventos, v1.0.5

# Requisitos Funcionais

**RF001: Cabeçalho** - *Ademir*

O sistema deve ter um cabeçalho fixado no topo de todas as páginas do site. Este cabeçalho deve permitir o acesso as seguintes páginas:

* Página inicial;
* Página de autenticação;
* Página de lista de eventos;
* Página de criação de evento.

Além de permitir que o usuário desfaça sua autenticação pelo cabeçalho.

**RF002: Autenticação do usuário** - *Matheus*

O sistema deve indicar o estado de autenticação do usuário através do cabeçalho. Ao usuário se autenticar o sistema deve redirecioná-lo para a página inicial.

**RF003: Página inicial** - *Ademir, Gustavo, Matheus, Vinicius*

O sistema deve oferecer uma página inicial contendo um motto e uma descrição do serviço oferecido. Esta página também deve permitir que o usuário crie um evento, caso ele esteja autenticado.

**RF004: Cadastro de uma conta de usuário** - *Matheus*

O sistema deve permitir que o usuário crie uma conta, tendo que informar os seguintes dados em sua criação:

* Nome;
* Sobrenome;
* Senha;
* Email;
* CPF;
* Número de celular.

**RF005: Email de uma conta de usuário** - *Matheus*

O sistema deve fazer a validação do email no cadastro e login de usuário, caso o email informado seja inválido, o sistema deve informar o usuário e impedir que o cadastro prossiga até que o erro tenha sido corrigido.

**RF006: CPF de uma conta de usuário** - *Matheus*

O sistema deve fazer a validação do CPF no cadastro de usuário, caso o CPF informado seja inválido ou não exista, o sistema deve informar o usuário e impedir que o cadastro prossiga até que o erro tenha sido corrigido.

**RF007: Número de celular de uma conta de usuário** - *Matheus*

O sistema deve fazer a validação do número de celular no cadastro de usuário, caso o número de celular informado seja inválido, o sistema deve informar o usuário e impedir que o cadastro prossiga até que o erro tenha sido corrigido.

**RF008: Máscaras dos dados de uma conta de usuário** - *Matheus*

O sistema deve utilizar as seguintes regras em relação as máscaras de dados no cadastro de usuário:

* Nome: Permitir apenas letras;
* Sobrenome: Permitir apenas letras;
* CPF: Permitir apenas números e limitado a 11 dígitos (hífen e pontos serão tratados pelo sistema).

**RF009: Máscara de senha de uma conta de usuário** - *Matheus*

O sistema deve mascarar os dados da senha no cadastro e login de usuário, não permitindo que seus dados fiquem visíveis.

**RF010: Confirmação de senha** - *Matheus*

O sistema deve pedir a confirmação da senha no cadastro de usuário, caso as senhas informadas sejam diferentes, o sistema deve informar o usuário e impedir que o cadastro prossiga até que o erro tenha sido corrigido.

**RF011: Controle de contas de usuário** - *Matheus*

O sistema deve verificar se não há conflito entre os dados informados no cadastro de usuário e os dados de contas já existentes, caso contenha conflito, o sistema deve informar o usuário e impedir que o cadastro prossiga até que o erro tenha sido corrigido.

**RF012: Verificação do email da conta de usuário** – *Ademir, Gustavo, Matheus, Vinicius*

O sistema deve enviar um email de verificação para confirmar que o email informado no cadastro de usuário pertence a pessoa. Ao seguir o link enviado a conta do usuário será ativada.

**RF013: Conteúdo do email de verificação de email** - *Ademir*

O email de verificação de email deve conter:

* Identificação do site;
* Mensagem informando que o usuário deve seguir o link para ativar sua conta;
* Link para confirmar a ativação da conta.

**RF014: Recuperação da senha** - *Ademir, Gustavo, Matheus, Vinicius*

O sistema deve permitir que o usuário recupere sua senha, sendo necessário que o usuário siga o link enviado para o seu email. Ao seguir o link enviado ao seu email o usuário pode informar uma nova senha para sua conta.

**RF015: Criação de um evento** - *Gustavo*

O sistema deve permitir a criação de um evento por usuários autenticados. Na criação de um evento o usuário deverá informar os seguintes dados:

* Nome;
* Data: dia, mês, ano;
* Local: estado, cidade, bairro, rua, número (opcional), complemento (opcional);
* Preço de entrada (opcional);
* Limite de acompanhantes por convidado (pode ser alterado individualmente);
* Idade mínima permitida (opcional);
* Traje (opcional);
* Descrição;
* Imagem ilustrativa (opcional);
* Cronograma;
* Anexos (opcional).

Todo evento terá uma lista de convidados que pode ser modificada a qualquer momento pelo usuário dono do evento.

**RF016: Preço de entrada de um evento** - *Gustavo*

O sistema deve permitir que o usuário defina um valor monetário ou um valor diverso como preço de entrada do seu evento.

**RF017: Máscaras dos dados de um evento** - *Gustavo*

O sistema deve utilizar as seguintes regras em relação as máscaras de dados na criação de um evento:

* Nome: Permitir apenas letras;
* Limite de acompanhantes por convidado: Permitir apenas números e limitar de 0 a 128;
* Idade mínima: Permitir apenas números e limitar de 1 a 128 anos.

**RF018: Finalização da criação de um evento** - *Gustavo*

Ao finalizar a criação de um evento o sistema deve redirecionar o usuário para a visão geral de seu evento.

**RF019: Link para visão geral do evento** - *Vinicius*

O sistema deve permitir que qualquer pessoa com o link do evento possa acessar a visão geral do evento. Esse link deve estar disponível para o criador do evento e para todos os convidados através o email enviado para cada um.

**RF020: Visão geral de um evento** - *Vinicius*

O sistema deve disponibilizar uma visão geral do evento, onde todos os dados do evento devem estar disponíveis para visualização em uma página. Esta página deve permitir o acesso a lista de convidados.

**RF021: Lista de convidados** - *Matheus*

O sistema deve permitir a visualização da lista de convidados de cada evento. Caso o usuário seja dono do evento ele poderá modificar esta lista, informando os seguintes dados para cada convidado:

* Nome;
* Email;
* Limite de acompanhantes (receberá valor definido na criação do evento caso não modificado).

**RF017: Máscaras dos dados de um convidado** - *Ademir, Gustavo, Matheus, Vinicius*

O sistema deve utilizar as seguintes regras em relação as máscaras de dados na de cada convidado do evento:

* Nome: Permitir apenas letras;
* Limite de acompanhantes: Permitir apenas números e limitar ao limite de acompanhantes do convidado.

**RF022: Envio de convites** - *Matheus*

O sistema deve enviar um convite por email para cada novo convidado adicionado na lista de convidados do evento.

**RF023: Conteúdo do email de convite** - *Ademir*

O email de convite enviado aos convidados adicionados na lista de convidados do evento deve conter:

* Identificação do site;
* Nome do usuário dono do evento;
* Link para visão geral do evento;
* Mensagem informando o convidado que foi... convidado ao evento;
* Link para status de presença;
* Mensagem informando que o link poderá ser acessado até a data de início do evento.

**RF024: Link para status de presença** - *Ademir*

O sistema deve construir um link único para cada convidado, este link deve redirecionar o convidado para a página de status de presença do evento e poderá ser usado até a data do evento.

**RF025: Status de presença** - *Matheus*

O sistema deve permitir acesso a página de convidados do evento a todo convidado através do link enviado ao seu email, nesta página o convidado pode confirmar ou recusar sua presença no evento. Caso o convidado tenha confirmado sua presença ele/ela poderá informar o número de acompanhantes confirmados além de receber um email.

**RF026: Conteúdo do email de presença confirmada** - *Ademir*

O email de presença confirmada enviado aos convidados que confirmaram sua presença no evento deve conter:

* Identificação do site;
* Mensagem informando que sua presença foi confirmada no evento;
* Mensagem informando que o número de convidados confirmados pode ser modificado até a data de início do evento;
* Link para a visão geral do evento.

**RF027: Cancelamento de um Evento** - *Gustavo*

O sistema deve permitir que o usuário cancele um evento, ao confirmar o cancelamento do evento o usuário será redirecionado para a página de lista de eventos e todos os convidados do evento serão notificados do cancelamento do evento por email.

**RF028: Conteúdo do email de cancelamento de um evento** - *Ademir*

O email de cancelamento de evento enviado aos convidados deve conter:

* Identificação do site;
* Link para a visão geral do evento;
* Mensagem informando que o evento foi cancelado;
* Motivo pelo qual o evento foi cancelado.

**RF029: Reenvio automático de convites** - *Turma*

O sistema deve enviar um convite por email toda semana para todos os convidados que estão com seu status de presença pendente.

**RF030: Lista de eventos** - *Vinicius*

O sistema deve disponibilizar uma lista com todos os eventos criados pelo usuário como uma página, onde cada evento deve estar resumido em nome, descrição e imagem do mesmo.

**RF031: Mudança do status de presença** - *Gustavo*

O sistema deve permitir que os convidados modifiquem seu status de presença e acompanhantes confirmados até a data do evento através do link de status de presença.

**RF032: Sugestões de provedores de email** – *Ademir, Gustavo*

O sistema deve oferecer sugestões de provedores de email em todos os campos de email, estas sugestões devem aparecer apenas quando o usuário digitar @.

**RF033: Impressão da lista de convidados** – *Gustavo*

O sistema deve permitir a impressão da lista de convidados confirmados. Nesta lista estarão os nomes e número de acompanhantes confirmados de todos os convidados confirmados.

# Requisitos Não Funcionais

**RNF001: Endereço de email** - *Ademir*

O sistema deve utilizar o endereço de email [projetomyevents@outlook.com](mailto:projetomyevents@outlook.com) para o envio de convites.

**RNF002: Design responsivo** - *Ademir*

O sistema deve apresentar uma interface responsiva, se comportando adequadamente independente do dispositivo em que ela será acessada.

**RNF003: Linguagens do sistema** - *Ademir, Gustavo, Matheus, Vinicius*

O sistema deve ser implementado em Java e Typescript, utilizando os frameworks Spring e Angular, respectivamente.

**RNF004: Sistema de banco de dados** - *Ademir*

O sistema deve utilizar o sistema de banco de dados PostgreSQL.