

Documento de Arquitetura de Software

1. Introdução

Uma vez terminada a elicitação de requisitos de um sistema, obtém-se a necessidade de se ter uma solução de como deve ser estruturado o software de modo que venha atender as restrições e funcionalidades que o software necessita. E esta solução se define em partes neste documento de arquitetura de software.

1.1 Finalidade

Este documento apresenta uma visão geral abrangente da arquitetura do sistema e utiliza uma série de visões arquiteturais diferentes para ilustrar os diversos aspectos do sistema. Sua intenção é capturar e transmitir as decisões significativas do ponto de vista da arquitetura que foram tomadas em relação ao sistema.

1.2 Escopo

Este Documento de Arquitetura de Software se refere a Gerência de Festas (PartyPlanning)

1.3 Definições, Acrônimos e Abreviações

Elicitação: Técnica de obtenção de dados junto aos usuários detentores das informações.

1.4 Referências

Documento de Requisitos

1.5 Visão Geral

Apos a definição da finalidade e escopo deste documento, abaixo vamos estar descrevendo a arquitetura escolhida que melhor soluciona os problemas e necessidade encontrados neste projeto. Assim vamos estar apresentando em 5 visões sendo elas casos de uso, lógica, processo, implantação e implementação.

2. Representação da Arquitetura

Este documento apresenta a arquitetura como uma série de visões: visão de casos de uso, visão lógica, visão de processos, visão de implantação e visão de implementação. Essas visões são apresentadas como modelos e utilizam a Linguagem Unificada de Modelagem (UML).

3. Metas e Restrições de Arquitetura

Existem algumas restrições de requisito e de sistema principais que têm uma relação significativa com a arquitetura. São elas:

Referencia	Descrição
RSEG1	Não permitir o acesso a usuários não logados no sistema.
RIHC1	Não é aceito nenhuma entrada que contenha caracteres especiais.
RSEG2	O software do sistema deve ser integrado com as redes sociais e publicar (com a devida autorização do usuário) nelas apenas o Local, Data e Nome do Evento não sendo permitido a distribuição das listas de convidados.
RDES1	O sistema não pode travar/encerrar o processo com frequência.
RSEG3	Ao se conectar com as redes sociais e for ser feita alguma publicação, sempre pedir permissão.
RSEG4	Pedir permissão apenas para postar no mural de sua conta social.
RSEG5	Não acessar as redes sociais sem a devida autorização.
RSEG6	Não compartilhar dados com as redes integradas.

4. Visão de Casos de Uso

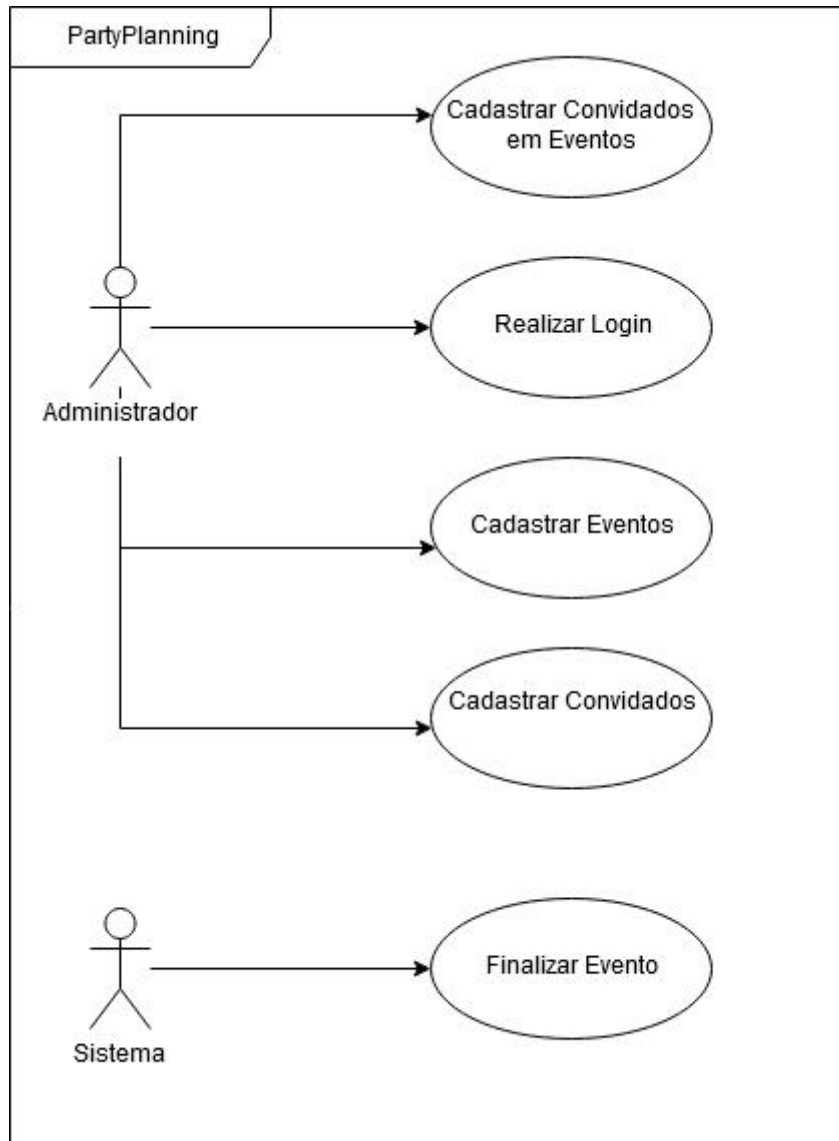
Uma descrição da visualização de casos de uso da arquitetura de software. A Visualização de Caso de Uso é uma entrada importante para a seleção do conjunto de cenários e/ou casos de uso que são o foco de uma iteração. Ela descreve o conjunto de cenários e/ou os casos de uso que representam alguma funcionalidade central e significativa. Também descreve o conjunto de cenários e/ou casos de uso que possuem cobertura arquitetural substancial (que exercita vários elementos de arquitetura) ou que enfatizam ou ilustram um determinado ponto complicado da arquitetura.

Os casos de uso do Gerência de Festas (PartyPlanning) são:

- Login
- Cadastro de novos Eventos,
- Cadastro de novos Convidados,

- Cadastro de Convidados em Eventos,
- Finalização de eventos,

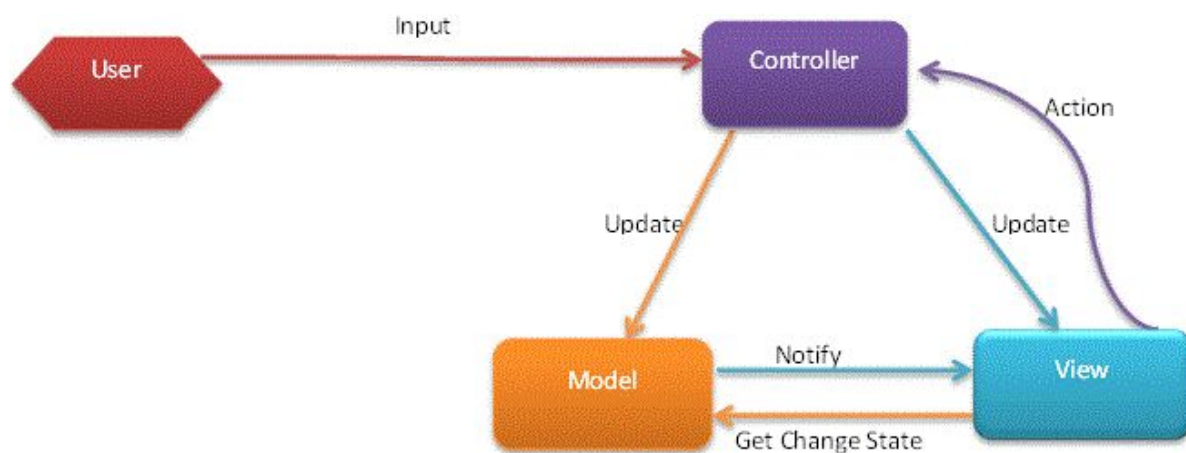
Esses casos de uso são iniciados pelo agente Administrador



5. Visão Lógica

Descreve as classes mais importantes, sua organização em pacotes e subsistemas de serviço, e a organização desses subsistemas em camadas.

5.1 Visão Geral



A visão geral do PartyPlanning é composta por 3 pacotes:

- **View:** Este não está preocupado em como a informação foi obtida ou onde ela foi obtida, apenas exibe a informação ao usuário.
- **Controller:** Este determina o fluxo de apresentação servindo como uma camada intermediária, ou seja, o intercessor entre o View e o Model.

- **Model:** Este é responsável por tudo o que a aplicação irá fazer. Modelando os dados e o comportamento do sistema, bem como preocupa com o armazenamento, manipulação e geração de dados, sendo um encapsulador de dados e de comportamento independente da apresentação (view).

6. Visão de Implantação

O sistema é construído na linguagem C# utilizando .NET Framework e MVC3, com o foco voltado para navegação Web.

7. Tamanho e Desempenho

O objetivo do PartyPlanning - Gerência de Eventos é fazer com que o máximo de usuários consigam acessar o app no menor tempo possível. todavia, os fatores de desempenho dependem do servidor em que o app estará hospedado.

8. Qualidade

Com a definição da arquitetura, houve uma independência proporcionada ao dividir em três partes principais a aplicação. O MVC também torna mais simples a manutenção do software, pois sua estrutura de arquivos torna fácil localizar trechos específicos de código, esse padrão arquitetural é a melhor solução para o modelo de aplicações web que propomos.