

---

**Interage!**

**Interage!**

# **Documento de Arquitetura de Software**

**Versão 5**

## Histórico da Revisão

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>
11/06/2017	1	Persistência dos dados	Cristopher, Maiza, Igor, Fellipe, Marcos, Alexandre
18/06/2017	2	Protótipos de tela	Cristopher, Maiza, Igor, Fellipe, Marcos, Alexandre
28/06/2017	3	Integração e protótipo da arquitetura	Cristopher, Maiza, Igor, Fellipe, Marcos, Alexandre
09/07/2017	4	Diagrama com visões da arquitetura	Cristopher, Maiza, Igor, Fellipe, Marcos, Alexandre
10/07/2017	5	Documento de arquitetura	Cristopher, Maiza, Igor, Fellipe, Marcos, Alexandre

## **Índice**

<b>Introdução</b>	5
Objetivo	5
Escopo	5
Definições, Acrônimos e Abreviações	5
Referências	5
Visão Geral	5
<b>Representação Arquitetural</b>	5
<b>Restrições e Metas Arquiteturais</b>	6
<b>Visão de Casos de Uso</b>	6
Logar	7
Recuperar de senha	8
Cadastrar de usuário	9
Agendar eventos	11
Adicionar atividade e sala de discussão	13
Pesquisar usuário	15
Atribuir perfil de Expositor a um Usuário Padrão	16
Pesquisar e ingressar em eventos	17
Participar de uma sala de discussão e interagir com Expositor de uma atividade	18
Enviar e visualizar feedbacks para um evento	19
<b>Visão Lógica</b>	20
Classes Significativas do Ponto de Vista da Arquitetura	22
Realizações de Casos de Uso	22
<b>Visão de Processos</b>	22
Logar	23
Cadastrar Usuário	24
Agendar evento	24
Criar sala de discussão	25
Pesquisar Usuário	27
Atribuir perfil de Expositor	28
Pesquisar Eventos	29
Ingressar em um evento	30
Entrar em uma sala de discussão	31

Interagir com Expositor	32
Enviar feedback para um evento	33
Visualizar feedback de um evento	34
<b>Visão da Implementação</b>	34
Visão Geral	34
Camadas	36
View	36
Controller	37
Model	37
<b>Visualização da Implantação</b>	38
<b>Visão de Dados</b>	38
<b>Tamanho e Desempenho</b>	40
<b>Qualidade</b>	40

---

## Documento de Arquitetura de Software

### 1. Introdução

O presente documento é composto pelo conjunto de visões demonstrativas do funcionamento lógico e estrutural do projeto Interage! visando tornar explícito o funcionamento de todas as suas partes e a função e meio de utilização de cada componente.

#### 1.1 Objetivo

Este documento fornece uma visão arquitetural abrangente do sistema, usando diversas visões de arquitetura para representar diferentes aspectos do sistema. Ele pretende capturar e transmitir as decisões arquiteturas significativas que foram tomadas em relação ao sistema.

#### 1.2 Escopo

Estão contemplados neste documento os padrões de software, componentes de software, plataformas de desenvolvimento, softwares de desenvolvimento, servidores de aplicação, servidores de banco de dados, frameworks e APIs.

São também descritos neste documento a descrição dos focos e sistemáticas arquiteturais, descrição das camadas de que é composto o modelo arquitetural, e requisitos de integrações.

Este documento limita-se a definir a arquitetura tecnológica da aplicação em si, sem adentrar em pormenores do funcionamento de ferramentas externas.

É também escopo deste documento orientar todo o pessoal técnico envolvido nas equipes de desenvolvimento.

#### 1.3 Definições, Acrônimos e Abreviações

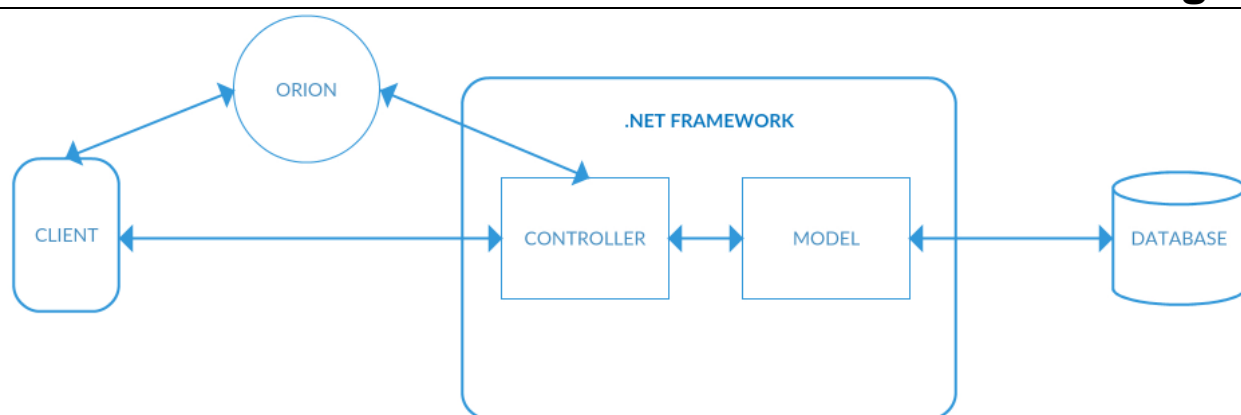
- **App** - aplicativo
- **API** - application programming interface
- **Context broker** - módulo do tipo Generic Enabler que recebe informações de fontes diferentes e repassa para um módulo consumidor dessas informações de maneira padronizada para que possam ser usadas na aplicação.

#### 1.4 Referências

#### 1.5 Visão Geral

Este documento contém outras 8 seções que contemplam a descrição da arquitetura de software do sistema atual e como ela é representada (seção 2); a descrição dos requisitos e objetivos do software que têm algum impacto sobre a arquitetura (seção 3); os casos de uso que representam funcionalidades significativas do sistema Interage! e seus respectivos protótipos de usabilidade e design (seção 4); uma apresentação das classes do sistema e descrições de suas responsabilidades, bem como alguns relacionamentos, operações e atributos de grande importância (seção 5); descrição da decomposição do sistema em processos (seção 6); uma descrição da estrutura geral do modelo de implementação e seus componentes (seção 7, diagrama de componentes); a descrição das configurações da rede física (nós e hardware) na qual o software é implantado e executado (seção 8); e, por fim, uma descrição da perspectiva de armazenamento de dados persistentes do sistema (seção 9).

## 2. Representação Arquitetural



### 3. Restrições e Metas Arquiteturais

O aplicativo é projetado apenas para usuários Android com versão mínima 4.4 KitKat.

A respeito das estratégias de implementação, usaremos o *NativeScript* e *Angular2* para o desenvolvimento do design visual e usabilidade e dos serviços de integração com Orion (context broker). Para desenvolvimento das APIs de conexão com o banco de dados MySQL, será usado um framework .NET chamado Entity.

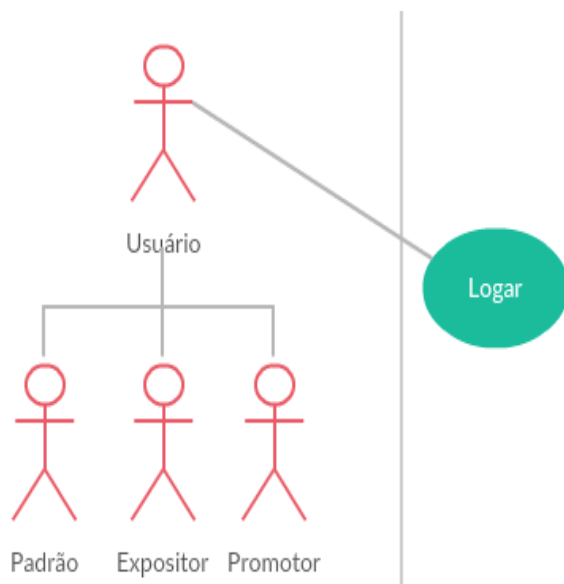
A estrutura da equipe segue o modelo SCRUM e está dividida entre Product Owner, Scrum Master, desenvolvedor mobile, desenvolvedor de back-end e analista de documentação.

### 4. Visão de Casos de Uso

Esta seção lista os casos de uso que representam funcionalidades significativas do sistema Interage! e seus respectivos protótipos de usabilidade e design, através dos protótipos de tela. Cada subseção diz respeito a um caso de uso.

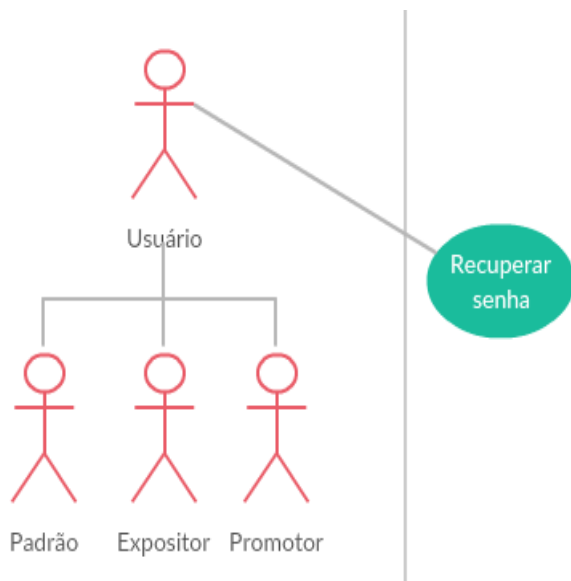
## 4.1 Logar

Este caso de uso representa o primeiro passo de qualquer usuário no Interage!. Assim que ele acessa a aplicação, caso já possua cadastro e queira prosseguir com as demais funcionalidade do seu perfil, ele precisa efetuar o login. O usuário precisa estar credenciado.



## 4.2 Recuperar de senha

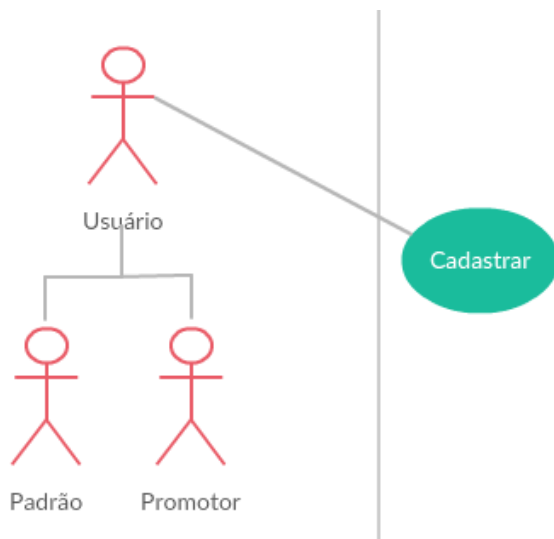
Este caso de uso ocorre quando o usuário deseja redefinir sua senha de acesso ao aplicativo. Ele deve apenas fornecer o e-mail cadastrado na sua conta.





## 4.3 Cadastrar de usuário

Caso de uso para o cadastro do usuário, tanto Promotor quanto Padrão. Para que o usuário tenha acesso às funções do Interage!, ele deve passar por este caso.



**Interage!**

« Cadastro Promotor

Nome completo da pessoa jurídica

E-mail

Senha

CNPJ

Endereço

Salvar e entrar

**Interage!**

« Cadastro usuário padrão

Nome completo

E-mail

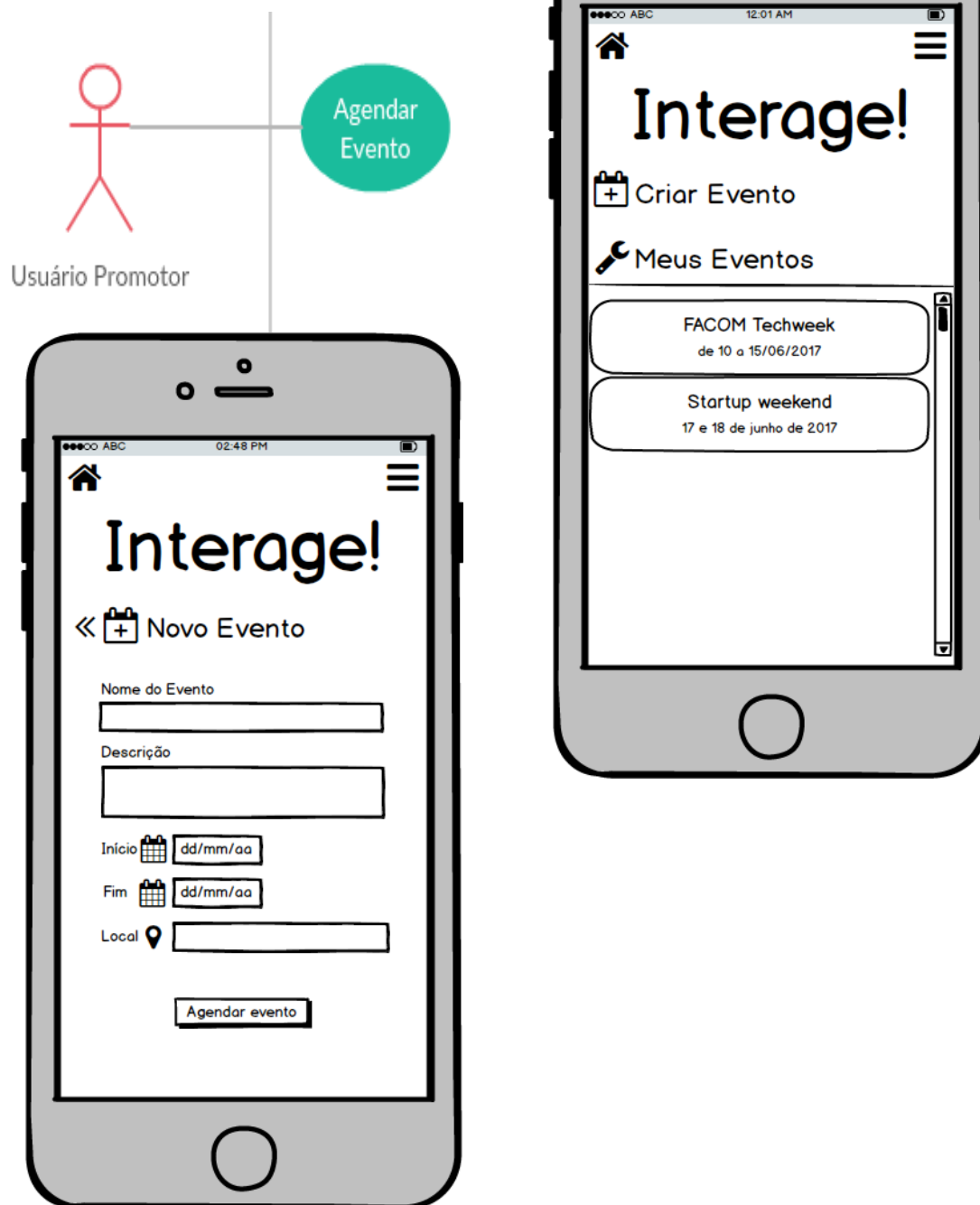
Senha

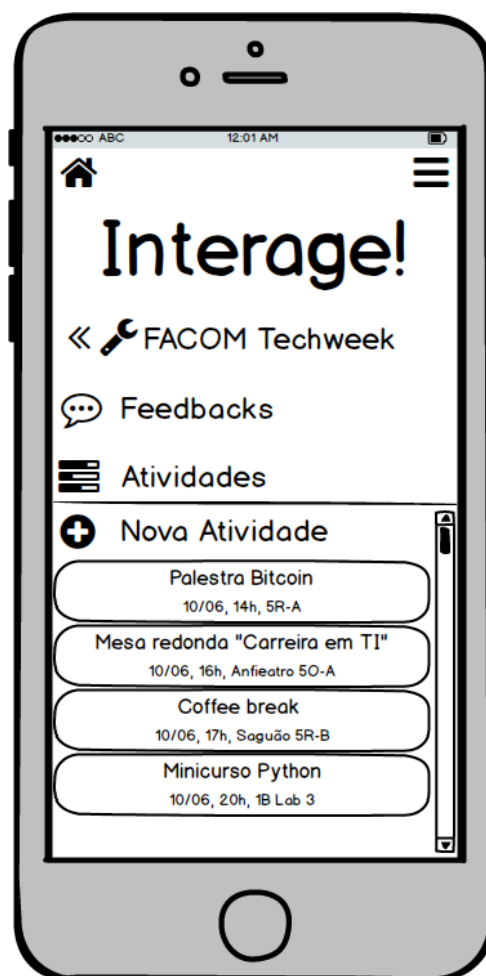
CPF

Salvar e entrar

## 4.4 Agendar eventos

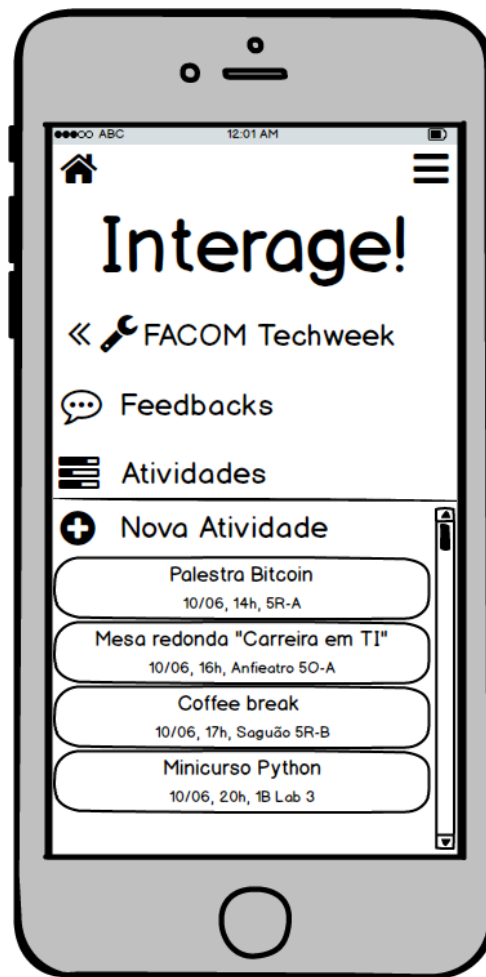
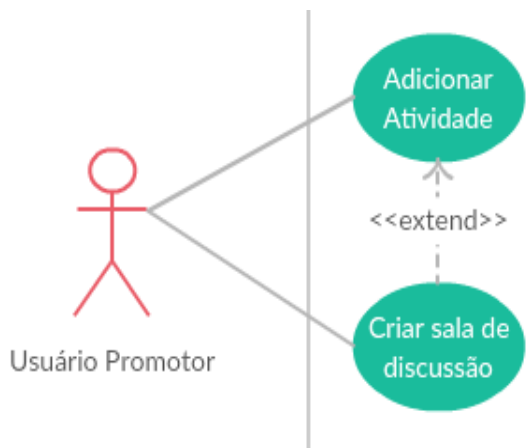
Este caso de uso ocorre quando um Usuário Promotor está logado no Interage! e deseja criar um novo evento que ficará sob sua gerência.





## 4.5 Adicionar atividade e sala de discussão

Em um evento criado pelo Usuário Promotor, ele pode adicionar atividades e escolher se essas terão ou não salas de discussão para interações com um Expositor.



The image shows a smartphone screen with the Interage! app interface. At the top, the status bar shows 'ABC' and '12:01 AM'. The app header features a home icon, the title 'Interage!', and a menu icon. Below the header, there is a back arrow and the text 'Nova Atividade'. The form includes several input fields: 'Nome da nova atividade', 'Descrição', 'Data' (with a calendar icon and 'dd/mm/aa' format), 'Início' and 'Fim' (with clock icons and 'hh:mm' format), 'Local' (with a location pin icon), and 'Expositor' (with a person icon and a search icon). There is also a checkbox for 'Sala de discussão' and an 'Adicionar' button at the bottom.

Nome da nova atividade

Descrição

Data dd/mm/aa

Início hh:mm Fim hh:mm

Local

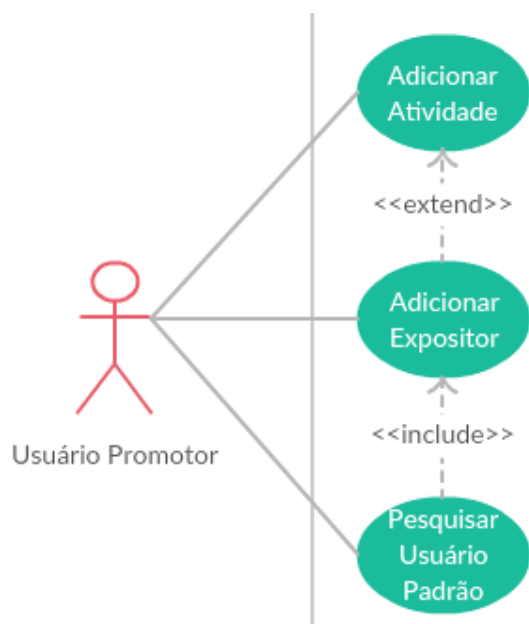
Expositor

☐ Sala de discussão

Adicionar

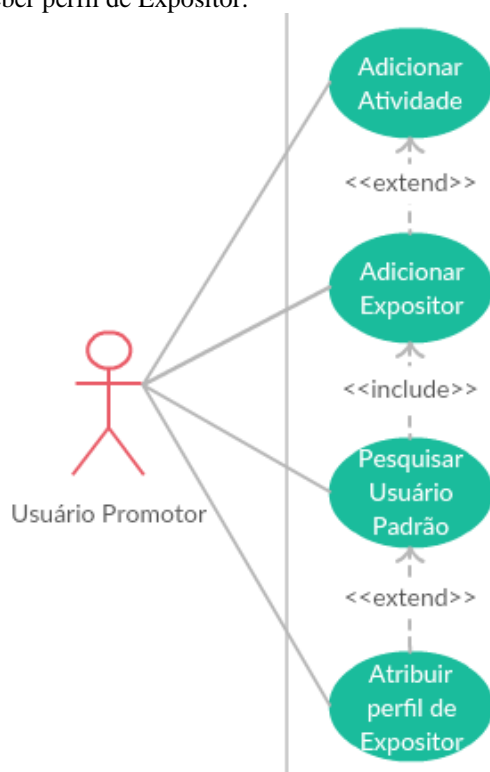
## 4.6 Pesquisar usuário

O Usuário Promotor pode pesquisar Usuários Padrão para, eventualmente, atribuí-los perfil de Expositor de uma atividade, por exemplo.



## 4.7 Atribuir perfil de Expositor a um Usuário Padrão

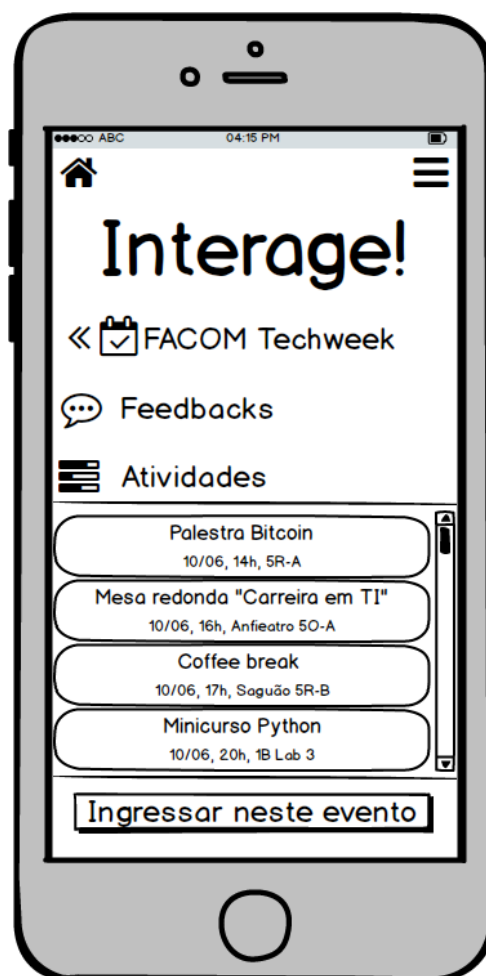
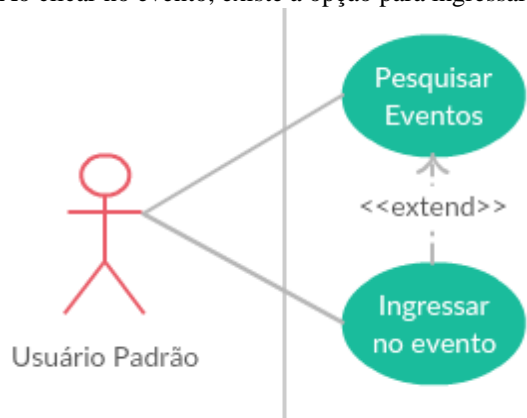
Neste caso de uso, um Usuário Promotor pode, dentro de uma atividade criada em um evento gerido por ele, pesquisar por um Usuário Padrão e selecioná-lo para receber perfil de Expositor.





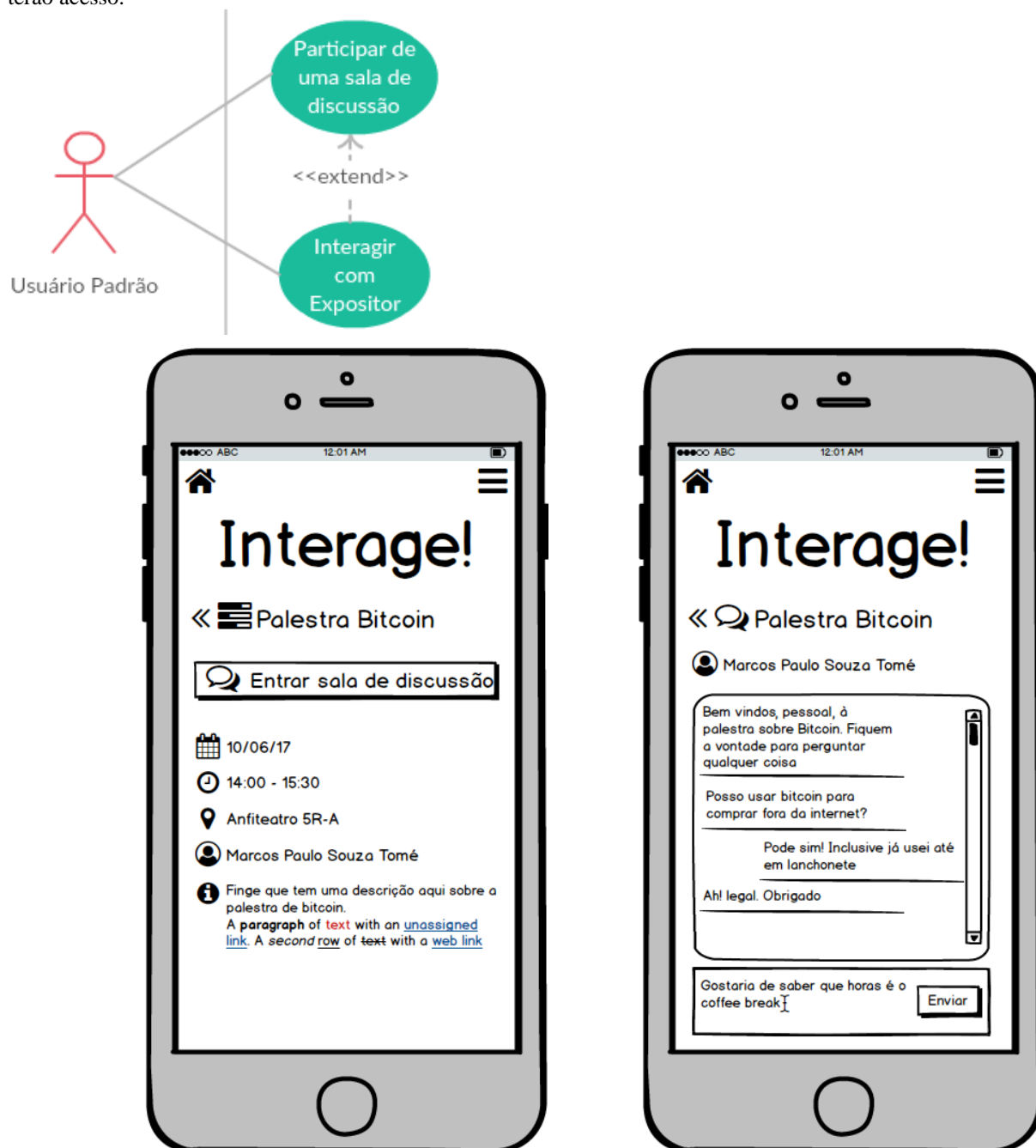
## 4.8 Pesquisar e ingressar em eventos

Neste caso de uso, o Usuário Padrão pode, pela *home* do aplicativo, pesquisar por eventos por nome e proximidade. Ao clicar no evento, existe a opção para ingressar no evento.



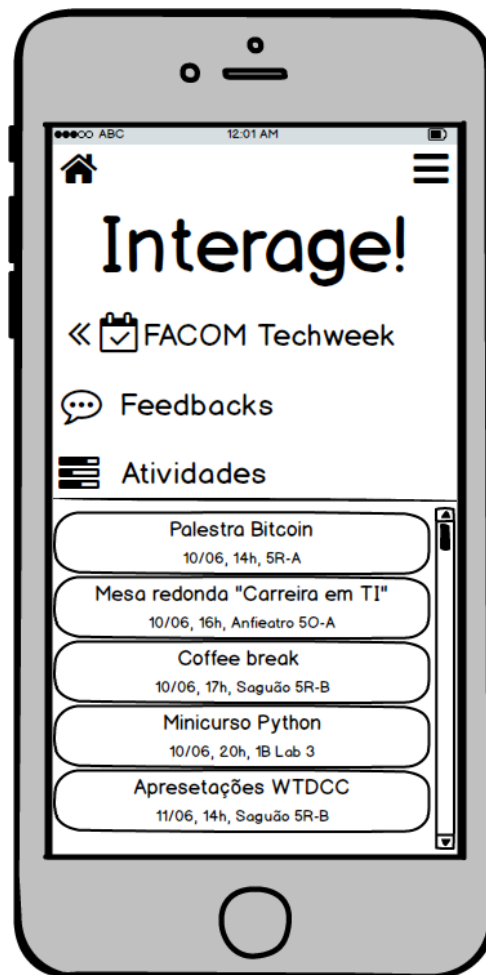
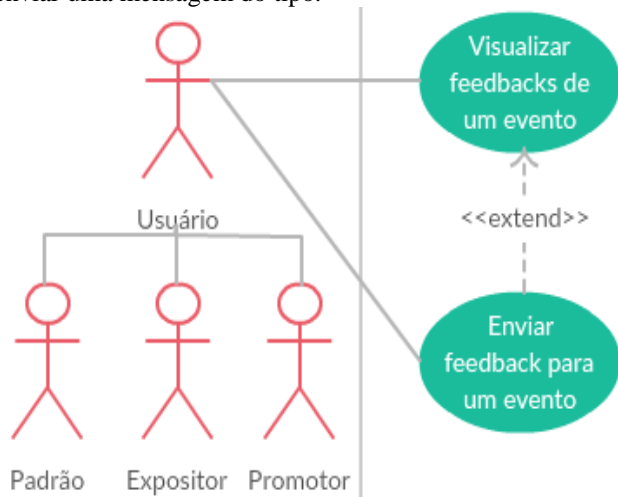
## 4.9 Participar de uma sala de discussão e interagir com Expositor de uma atividade

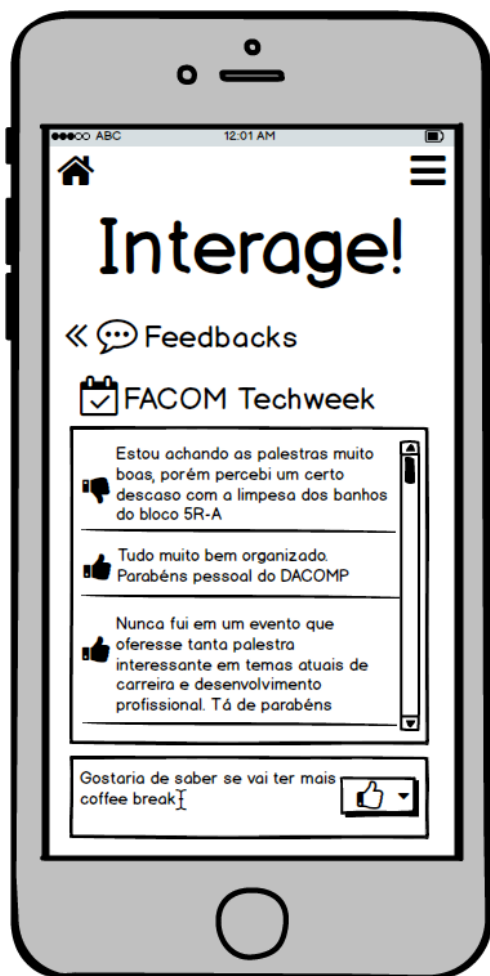
Ao ingressar em um evento, o Usuário Padrão tem acesso às atividades daquele evento. Dependendo da natureza da atividade, ela pode ou não ter uma sala de discussão para eventuais interações com o Expositor dessa atividade. O Usuário padrão pode, então, participar dessas salas e enviar mensagens para ela, onde todos os participantes da sala terão acesso.



## 4.10 Enviar e visualizar feedbacks para um evento

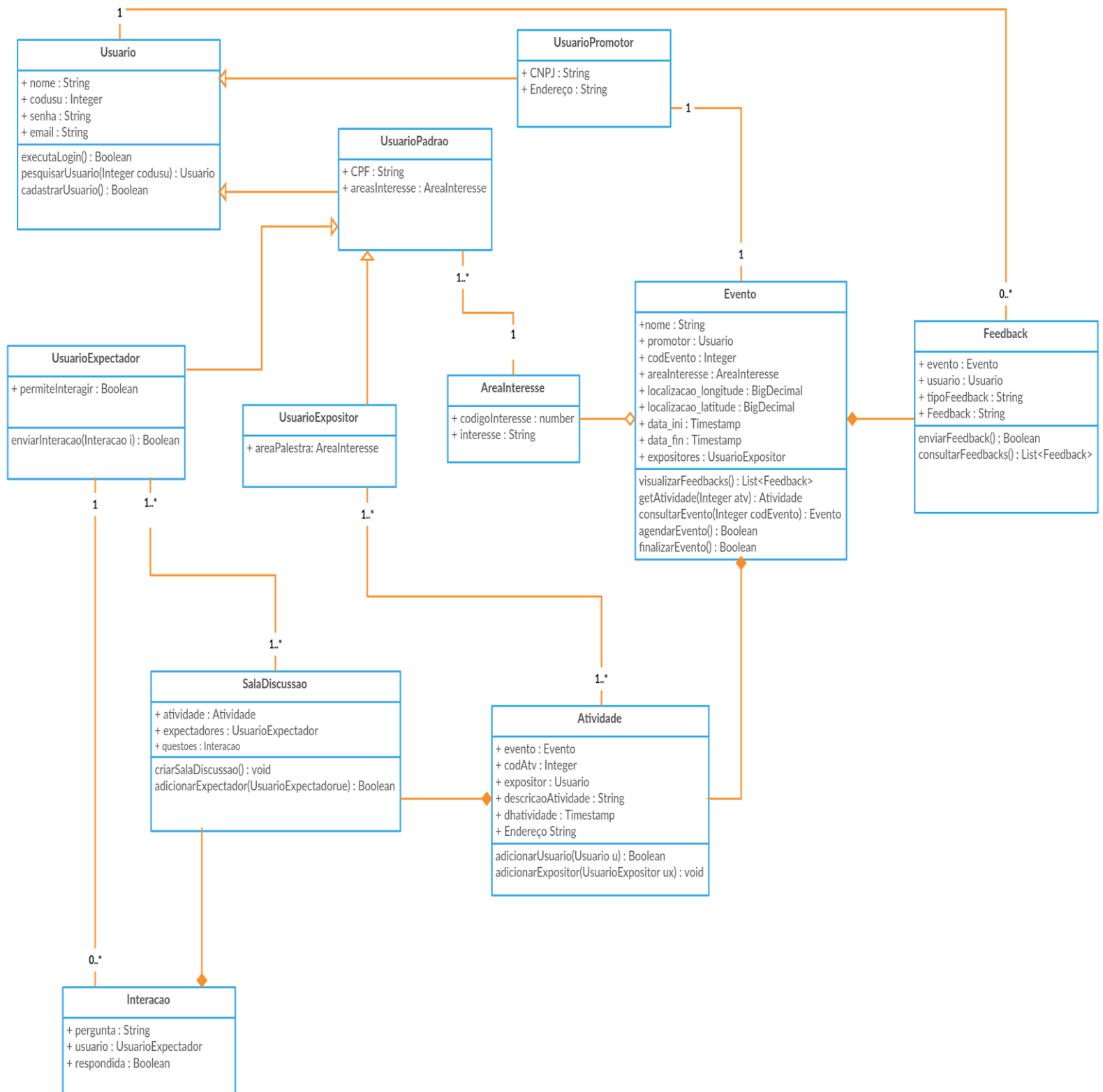
Neste caso de uso, os usuários podem, através da tela inicial do evento, entrar na seção de Feedbacks dele e enviar uma mensagem do tipo.





## 5. Visão Lógica

Esta seção demonstra o aspecto lógico do modelo de arquitetura através da divisão de classes do sistema Interage!, assim como suas responsabilidades, relacionamentos, operações e atributos.



## 5.1 Classes Significativas do Ponto de Vista da Arquitetura

Classes do sistema:

Usuario: é uma superclasse que representa um usuário do sistema, este ao efetuar o cadastro terá como opções se cadastrar como usuário padrão ou usuário promotor, que são representados pelas subclasses UsuarioPadrao e UsuarioPromotor, respectivamente.

UsuarioPromotor: essa classe representa um usuário promotor, este usuário tem permissões para agendar um evento e atribuir a um usuário padrão o perfil de expositor.

UsuarioPadrao: classe que representa um usuário padrão, este usuário provê informações sobre suas áreas de interesse. Ingresa em um evento podendo ter perfil de espectador ou expositor, representados pelas classes UsuarioExpectador e UsuarioExpositor, respectivamente.

UsuarioExpectador: representa um usuário espectador, este usuário pode interagir com a sala de discussão da qual está participando. A interação e a sala de discussão são representados pelas classes Interacao e SalaDiscussao, respectivamente.

UsuarioExpositor: representa um usuário expositor, este perfil de usuário deve ser atribuído pelo usuário promotor ao usuário padrão. O expositor participa de determinada atividade do evento. A atividade e o evento são representados pelas classes Atividades e Evento, respectivamente.

SalaDiscussao: representa uma sala de discussão de determinada atividade do evento, criada pelo usuário promotor.

Atividade: representa uma atividade do evento.

Evento: representa um evento cadastrado pelo usuário promotor.

Feedback: representação uma avaliação feita pelo usuário padrão a respeito do evento.

## 5.2 Realizações de Casos de Uso

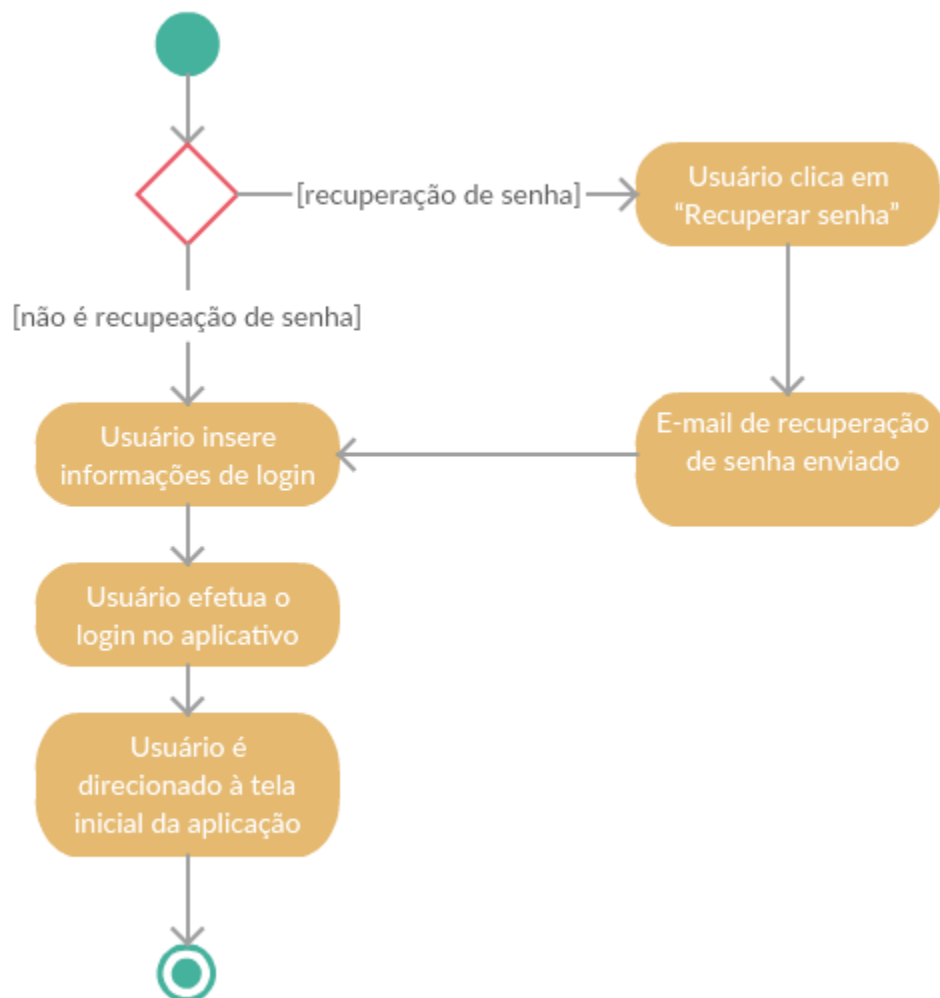
Com base nos seguintes casos de uso: UC - Interagir com Expositor e UC - Participar de sala de discussão. Segue um breve funcionamento do sistema com base nos casos de uso selecionados.

O usuário padrão ao ingressar no evento de seu interesse, busca por atividades desse evento e em seguida escolhe a sala de discussão vinculada a atividade que deseja participar. Ao inserir na sala como espectador, o usuário tem a opção de interagir com o usuário expositor, enviando a ele perguntas sobre a atividade discutida na sala.

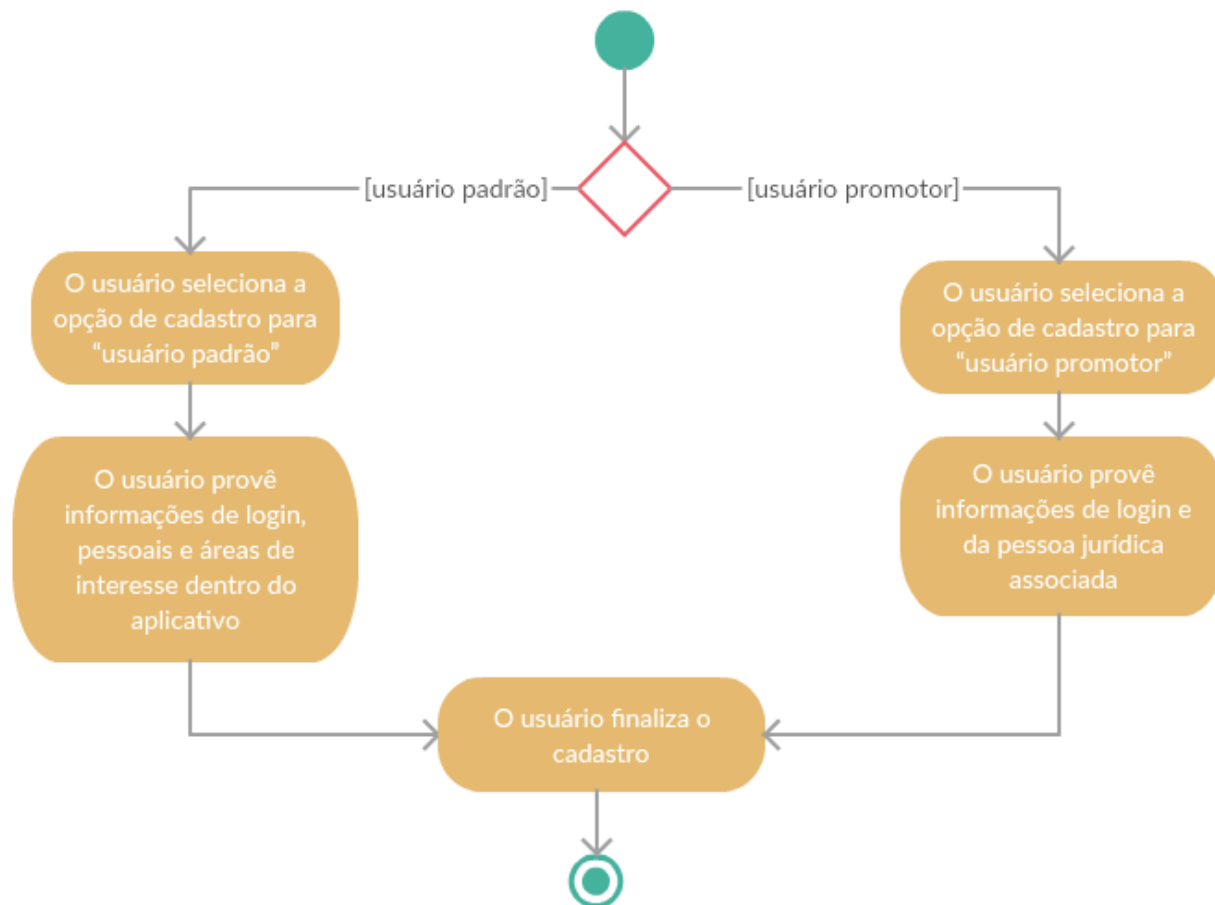
## 6. Visão de Processos

Esta seção descreve a decomposição do sistema em diagramas de atividades. Cada subseção abaixo identifica um diagrama de atividade correspondente aos casos de uso (funcionalidade) do Interage!

## 6.1 Logar

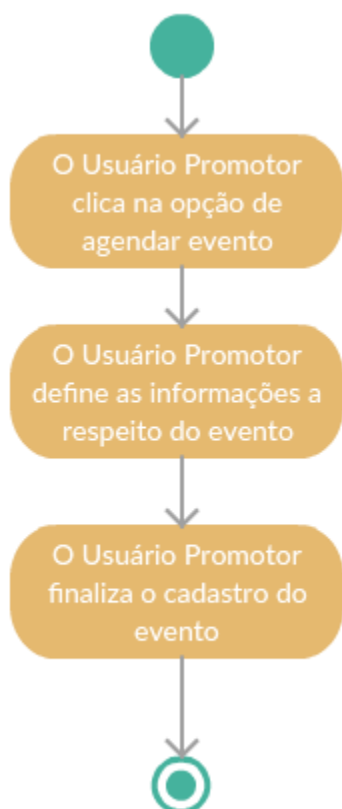


## 6.2 Cadastrar Usuário



## 6.3 Agendar evento

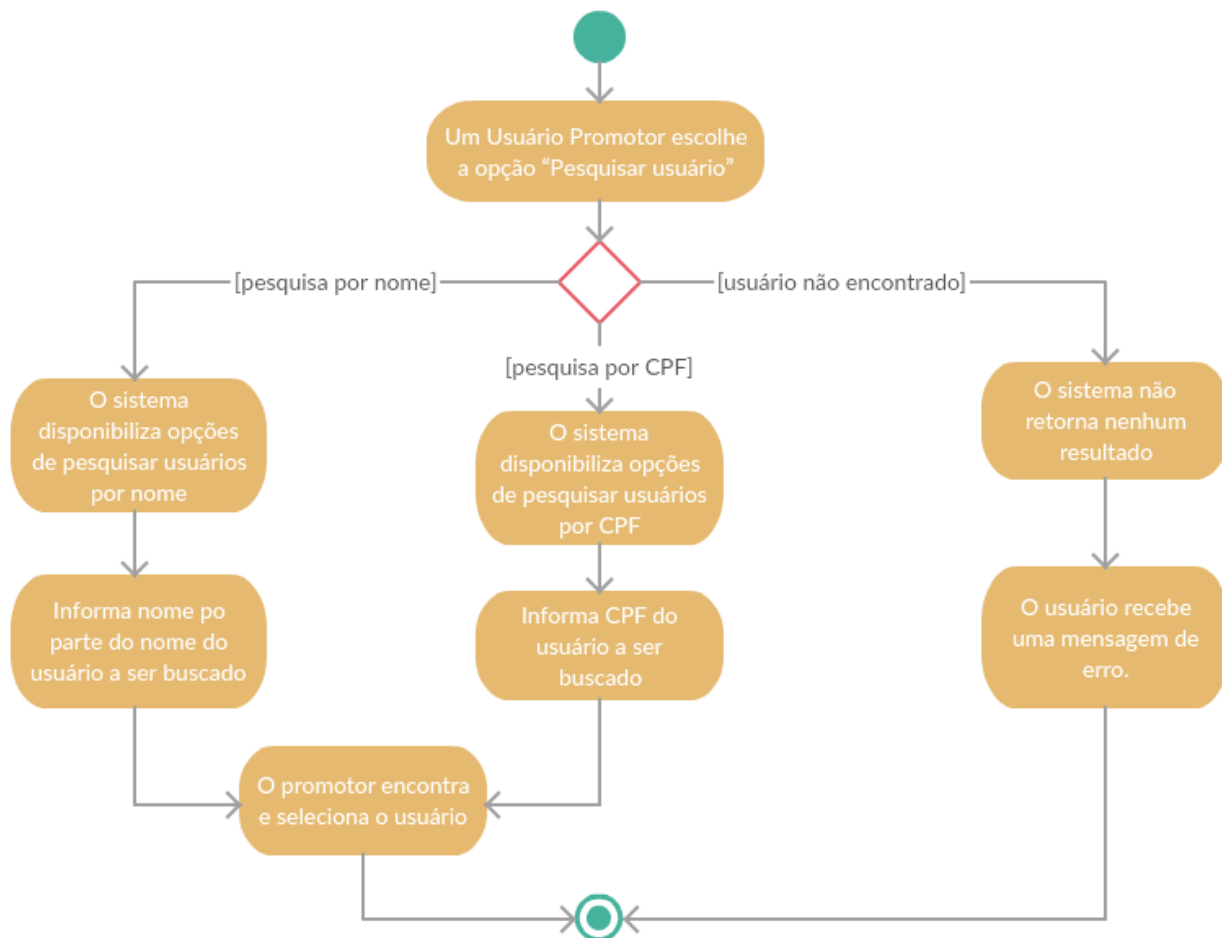




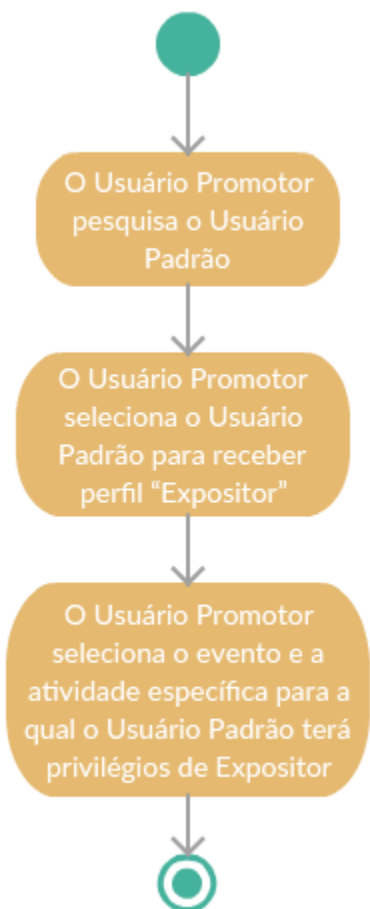
## 6.4 Criar sala de discussão



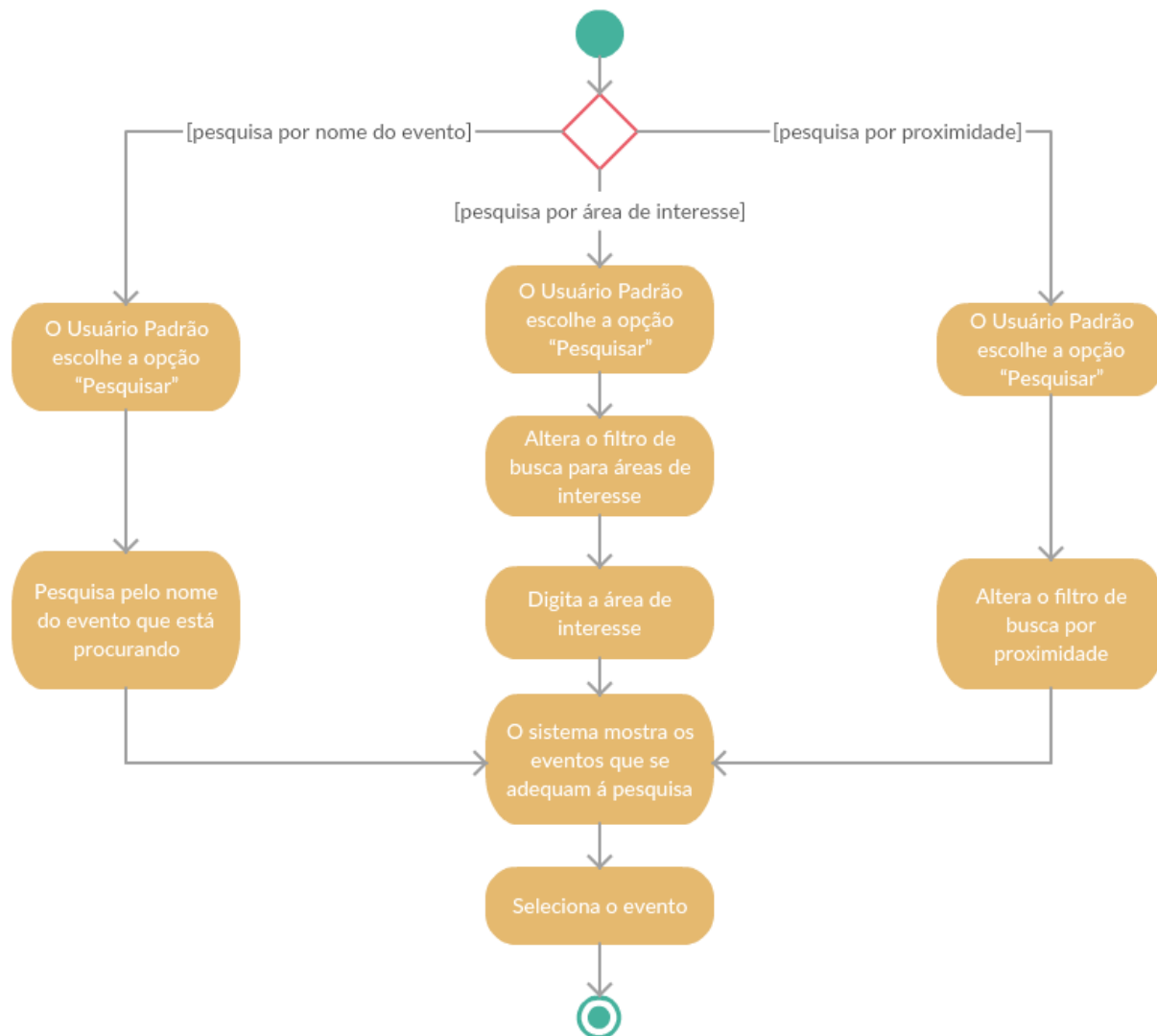
## 6.5 Pesquisar Usuário



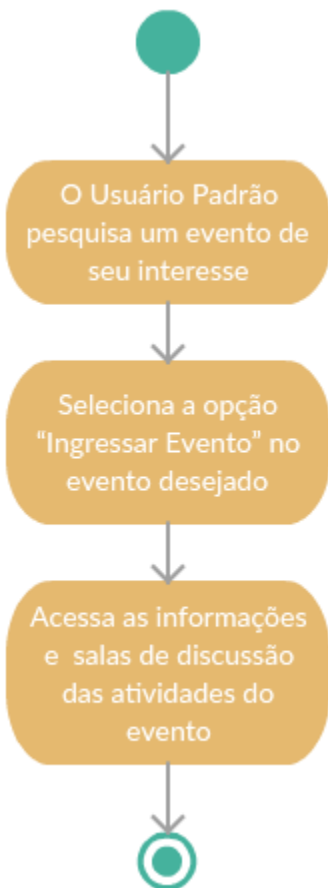
## 6.6 Atribuir perfil de Expositor



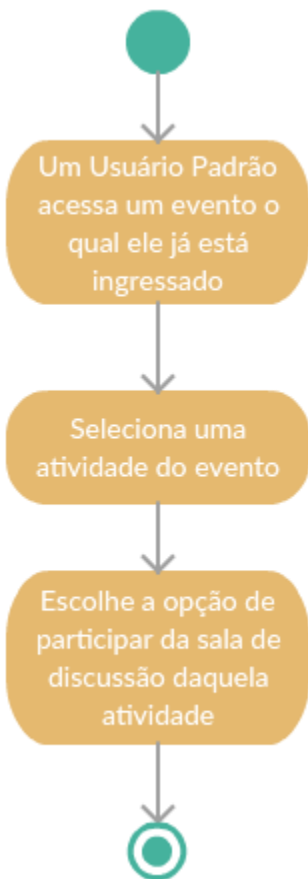
6.7 Pesquisar Eventos



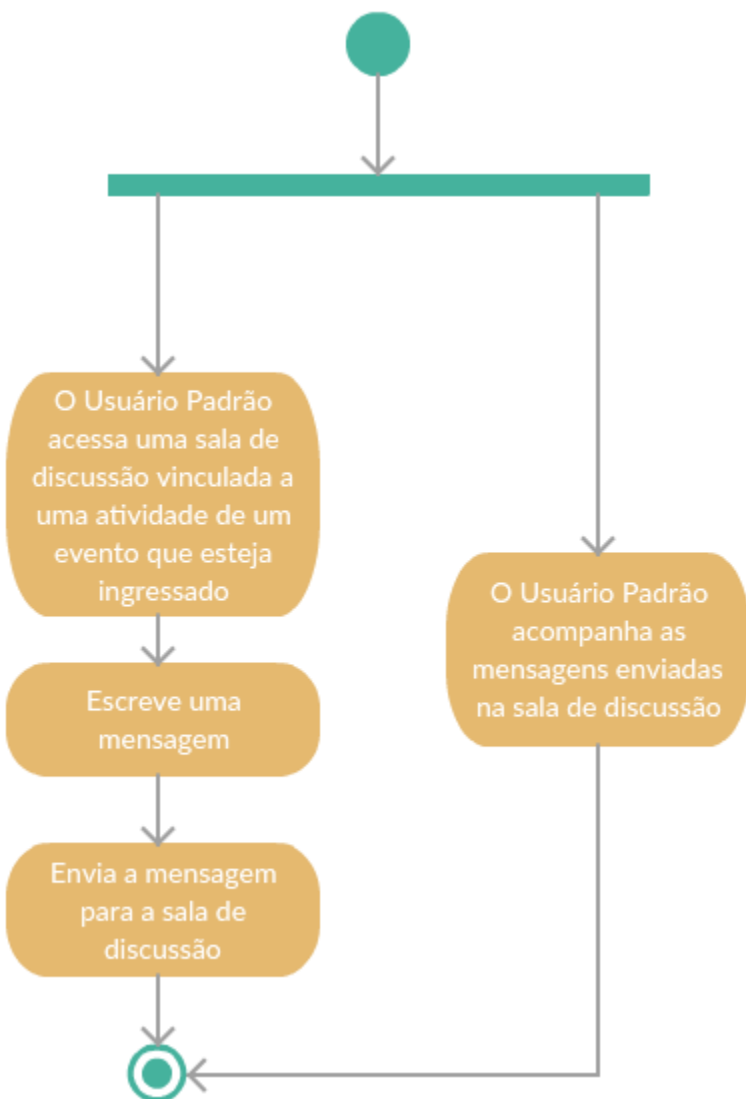
## 6.8 Ingressar em um evento



## 6.9 Entrar em uma sala de discussão

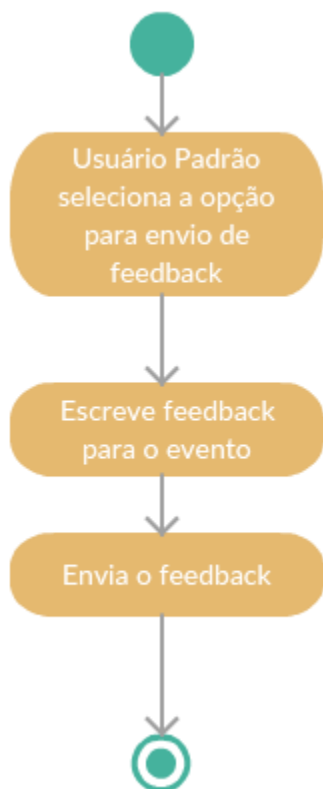


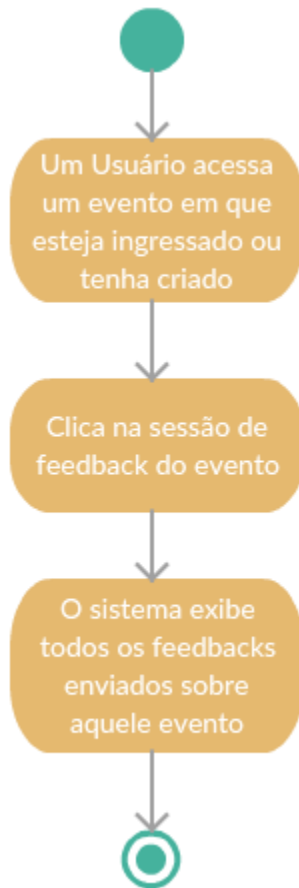
## 6.10 Interagir com Expositor



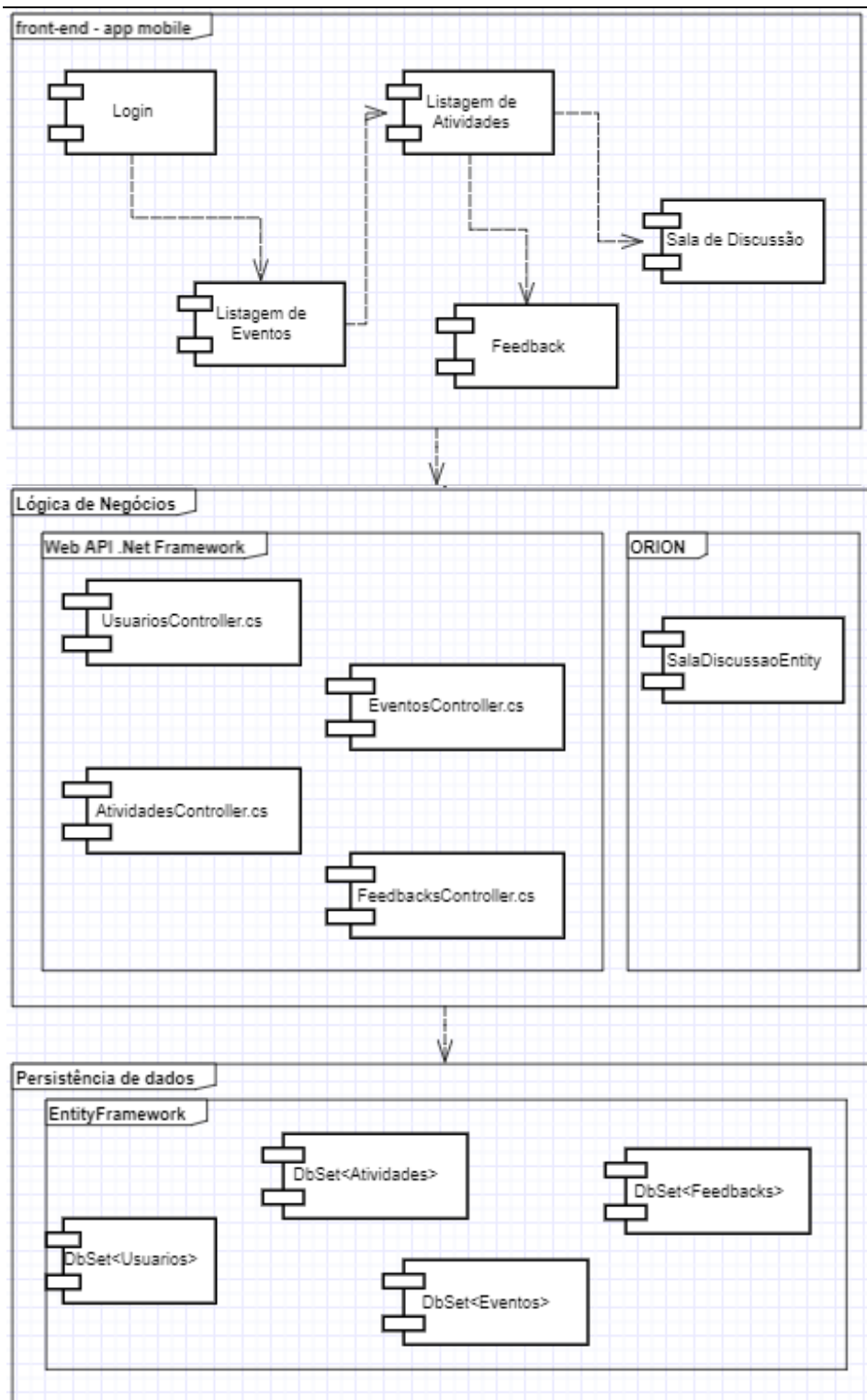


## 6.11 Enviar feedback para um evento



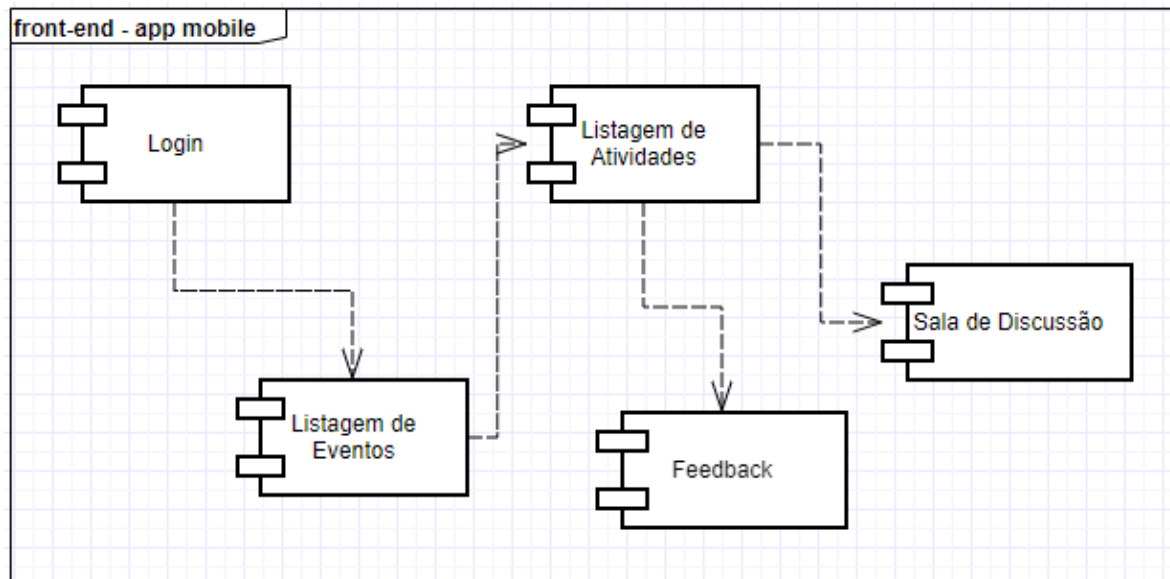
**6.12 Visualizar feedback de um evento****7. Visão da Implementação****7.1 Visão Geral**

A implementação do Interage! segue o padrão arquitetural MVC (Model, View, Controller), que divide a aplicação em 3 camada: o Model, que corresponde ao modelo de dados e as regras de acesso aos mesmos; a View, que refere-se à interface entre o usuário e aplicação; e o Controller, que faz uma ponte entre a View e o Model, pois recebe uma requisição do usuário (que envia através da View), e faz uma consulta ao Model e trata os dados (ou atualiza o modelo de dados com as informações recebidas) e retorna uma resposta para a camada de visualização.



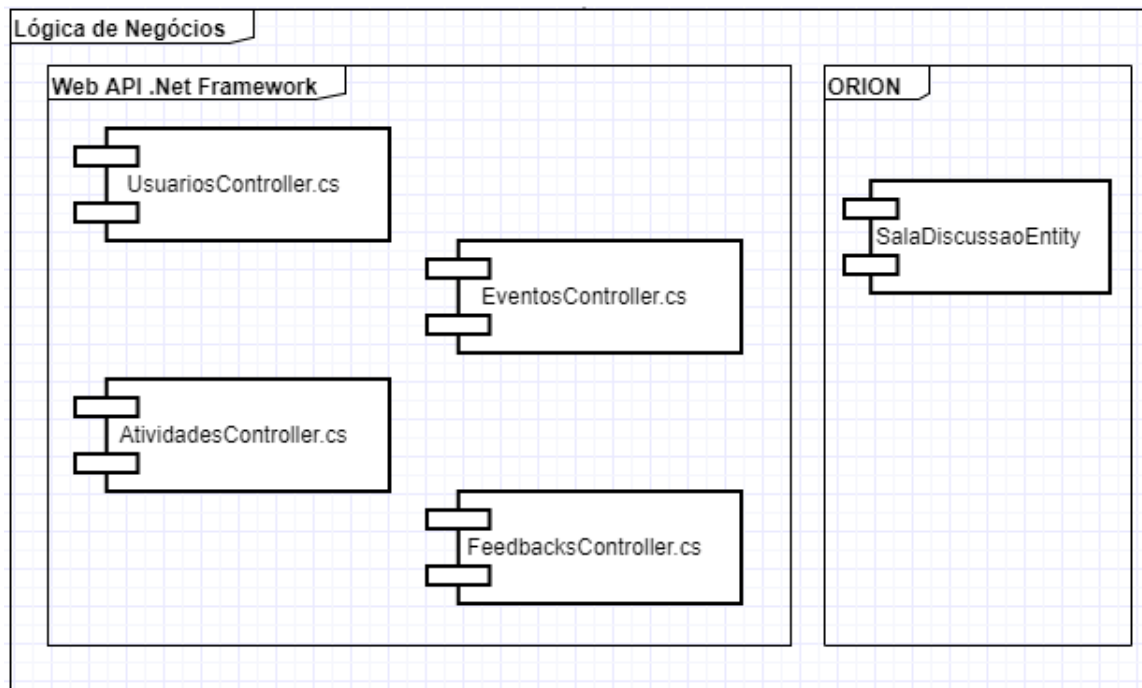
## 7.2 Camadas

### 7.2.1 View



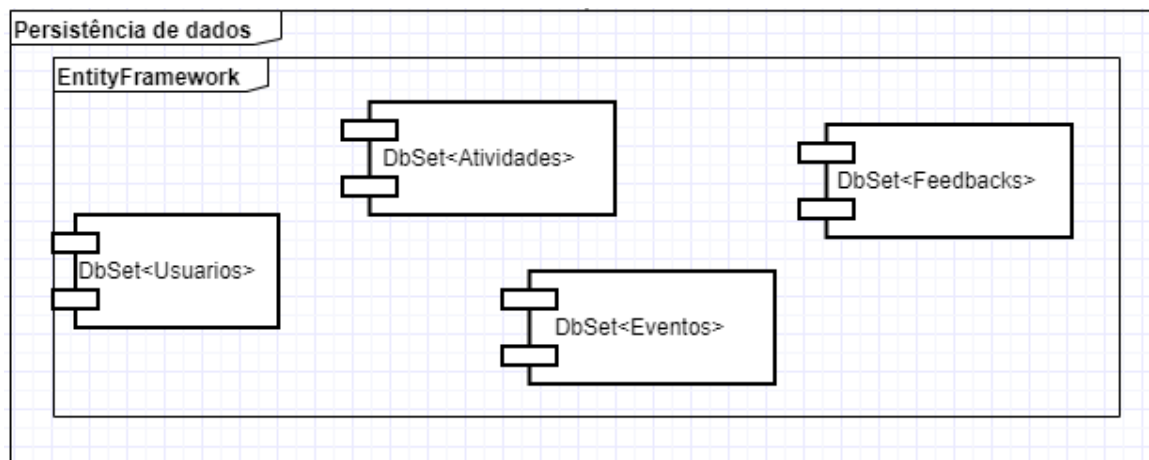
Na camada de visualização temos as telas que fazem a interface com os usuários: Login, Eventos, Atividades, Feedback e Sala de Discussão.

## 7.2.2 Controller



Na camada de Controller, temos os controllers implementados em .Net, responsáveis principalmente pelas operações de CRUD e autenticação, além do Orion, que é responsável por receber as interações dos usuários em uma sala de discussão, e enviar uma notificação para todos os membros da respectiva sala.

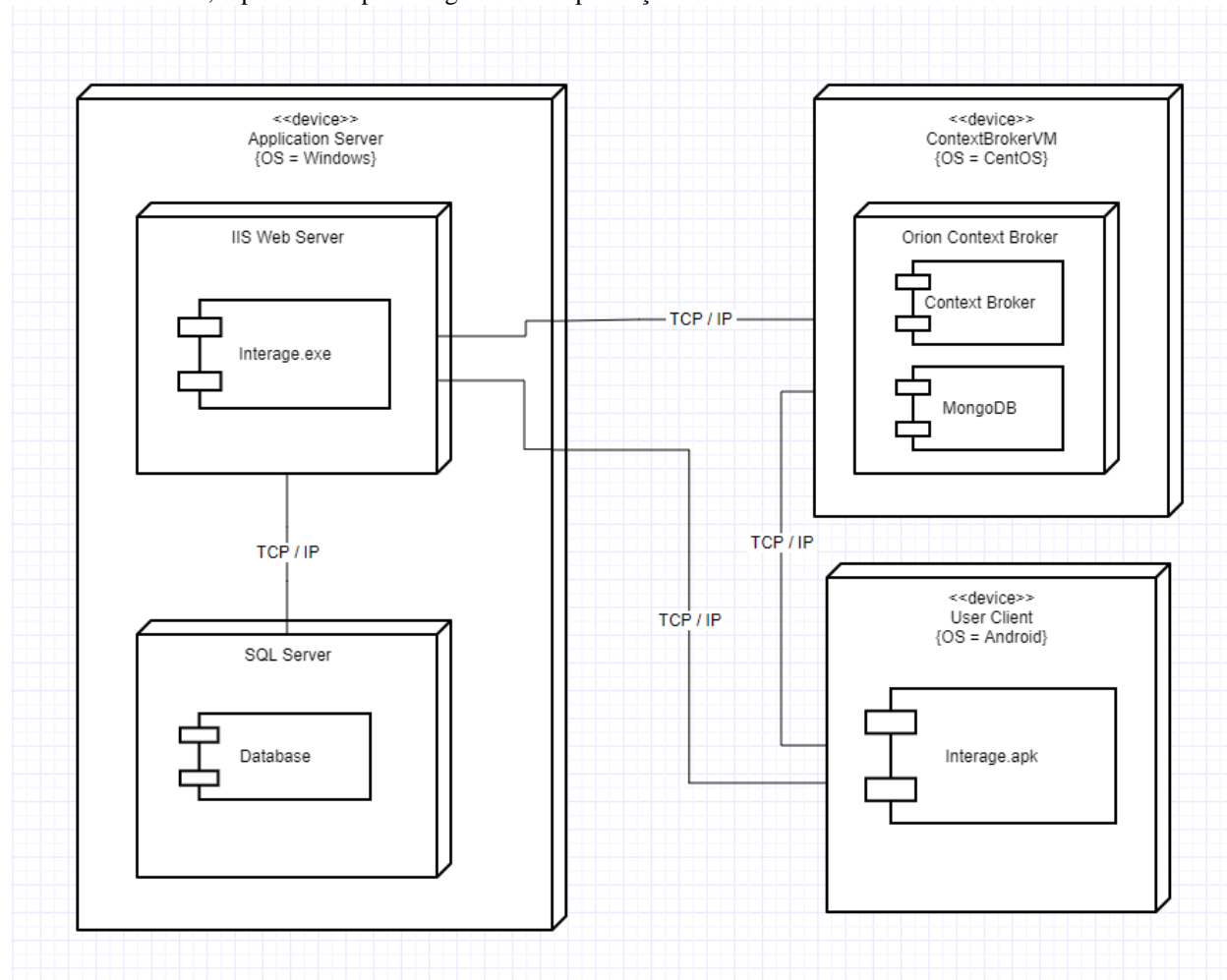
## 7.2.3 Model



Por fim, temos a representação macro do modelo de dados, com os DbSet que fazem o mapeamento para as tabelas no banco de dados.

## 8. Visualização da Implantação

Esta seção descreve os aspectos físicos da rede física na qual o software será implantado e executado, bem como suas interconexões, representado pelo diagrama de implantação.

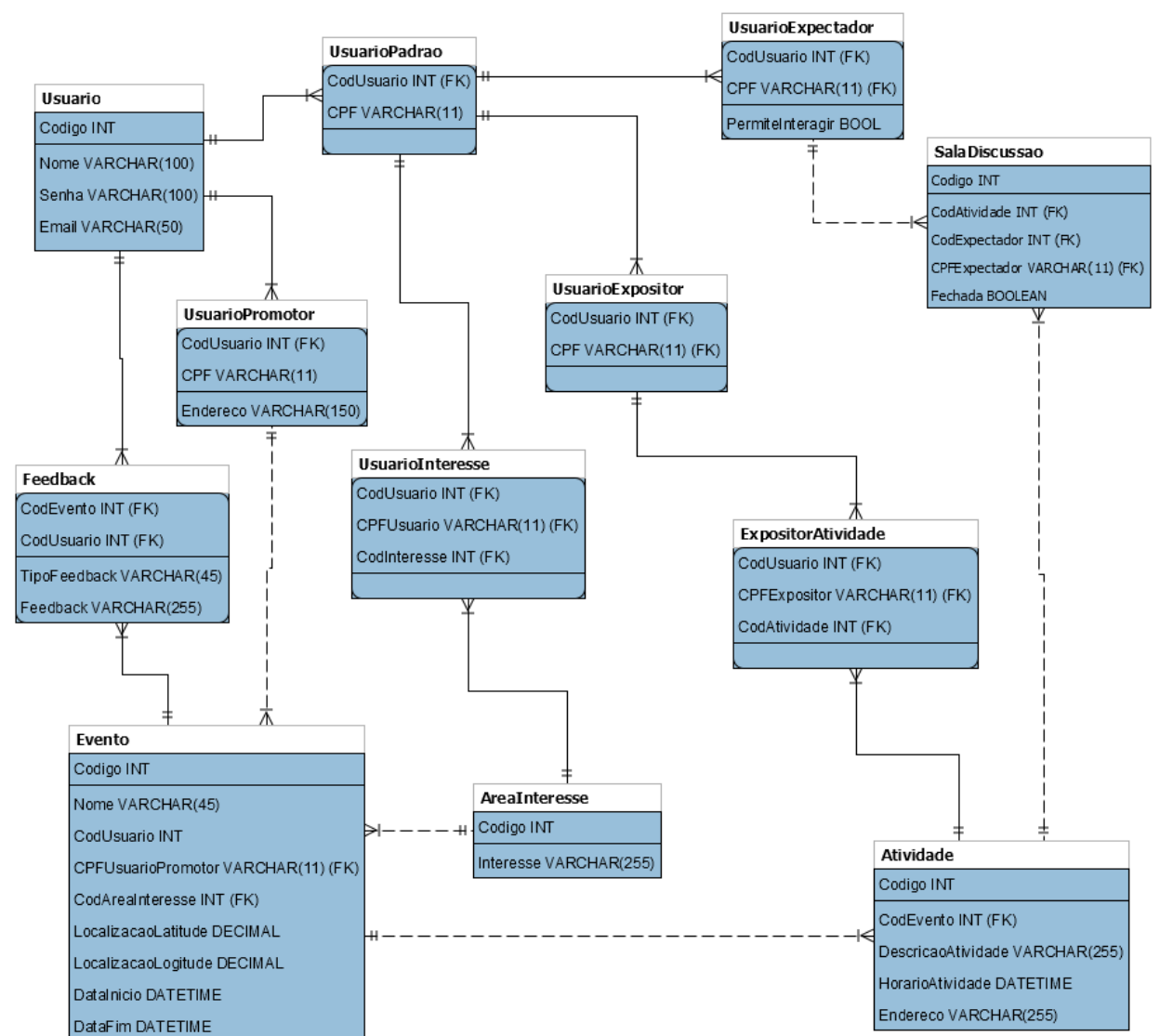


## 9. Visão de Dados

O aplicativo utilizar-se-á do context broker do Orion a fim de fazer a comunicação entre os diversos inputs fornecidos pelos usuários e a plataforma do aplicativo. O exemplo abaixo ilustra o caso de disponibilização de uma pergunta em uma sala de discussão.

```
{
  "contextElements": [
    {
      "type": "SalaDiscussao",
      "isPattern": "false",
      "id": "sala1",
      "attributes": [
        {
          "name": "codUsuario",
          "type": "Integer",
          "value": "2145"
        },
        {
          "name": "cpfUsuario",
          "type": "String",
          "value": "12332122234"
        },
        {
          "name": "pergunta",
          "type": "String",
          "value": "Pergunta?"
        },
        {
          "name": "flagRespondida",
          "type": "Boolean",
          "value": "true"
        }
      ]
    }
  ],
  "updateAction": "APPEND"
}
```

Os dados que serão persistidos corresponderão àqueles referentes aos usuários cadastrados, seus eventos e áreas de interesse e às identificações de elementos do evento como Salas de Discussão, Atividades, Questionamentos e Feedbacks disponibilizados por usuários.



## 10. Tamanho e Desempenho

O software, conforme projetado, tem como objetivo 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 esteja hospedado. A quantidade de usuário simultâneos deverá ser abaixo de 10.000.

## 11. Qualidade

N/A.