**INSTITUTO FEDERAL DO PARANÁ**

**LEONARDO QUEIROGA RAMIREZ**

**MAICKEL PIRCHINER DA COSTA**

**APLICATIVO PARA ORGANIZAÇÃO DE PARTIDAS ESPORTIVAS JOGA JUNTO**

FOZ DO IGUAÇU, PR

2019

**LEONARDO QUEIROGA RAMIREZ**

**MAICKEL PIRCHINER DA COSTA**

**APLICATIVO PARA ORGANIZAÇÃO DE PARTIDAS ESPORTIVAS JOGA JUNTO**

Documentação do Projeto Integrador apresentado ao curso Tecnólogo em Análise e Desenvolvimento de Sistemas como requisito parcial de avaliação.

Orientadores:

Marcela Turim Koschevic

Nelson Nunes André Junior

FOZ DO IGUAÇU, PR

2019

**FOLHA DE APROVAÇÃO**

LEONARDO QUEIROGA RAMIREZ

MAICKEL PIRCHINER DA COSTA

APLICATIVO PARA ORGANIZAÇÃO DE PARTIDAS ESPORTIVAS JOGA JUNTO

Documentação do Projeto Integrador apresentado como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas do Instituto Federal do Paraná, aprovada pela seguinte banca examinadora:

Orientador(a): Profª. Ma. Marcela Turim Koschevic

Colegiado de Informática, IFPR

Orientador(a): Nome completo do Orientador 2, com titulação

Colegiado de Informática, IFPR

Orientador(a): Nome completo do Orientador 3, com titulação

Colegiado de Informática, IFPR

Membro avaliador: Nome completo do Membro Avaliador 1, com titulação

Colegiado de {nome do colegiado}, IFPR

Foz do Iguaçu, XX de Junho de 2019

**RESUMO**

Elemento obrigatório, que evidencia os pontos mais importantes do trabalho, apresentando inclusive os resultados obtidos. Deve ser digitado em parágrafo único, com espaçamento simples, contendo de 150 a 500 palavras (teses, dissertações eoutros), e de 100 a 250 palavras (artigos de periódicos). Utilizar a terceira pessoa do singular mantendo o verbo na voz ativa. O termo RESUMO deve ser escrito em letras maiúsculas, em negrito e centralizado. Devem ser indicadas de três a cinco palavras-chave, separadas entre si por ponto, que representem o assunto abordado no trabalho. O termo Palavras-chave deve ser escrito em letras minúsculas, em negrito, com a primeira letra em maiúscula. Deve ser deixada uma linha em branco entre a palavra RESUMO.

**Palavras-chave:** Termo 1. Termo 2. Termo 3.

**ABSTRACT**

Elemento obrigatório para teses e dissertações, escrito, preferencialmente, na língua inglesa. Deve ser digitado em folha separada, da mesma forma e contendo as mesmas informações do resumo apresentado na língua portuguesa, inclusive as palavras-chave. O termo ABSTRACT deve ser escrito em letras maiúsculas, em negrito e centralizado. Devem ser indicadas de três a cinco palavras-chave, separadas entre si por ponto, que representem o assunto abordado no trabalho.

**Key words:** Termo 1. Termo 2. Termo 3.

**LISTA DE ILUSTRAÇÕES**

[Figura 1 - APLICATIVO CHEGA + 11](#_Toc7608568)

[Figura 2 - APLICATIVO DONO DA BOLA 12](#_Toc7608569)

[Figura 3 - aplicativo peladeiros i 12](#_Toc7608570)

[Figura 4 - aplicativo peladeiros ii 13](#_Toc7608571)

[Figura 5 - Diagrama de Casos de Uso 19](#_Toc7608572)

[Figura 6 - Diagrama de Classes Conceitual 20](#_Toc7608573)

[Figura 7 - Diagrama de Classes de Implementação 21](#_Toc7608574)

[Figura 8 - Diagrama de Entidade-Relacionamento 22](#_Toc7608575)

[Figura 9 - Diagrama de Atividades 23](#_Toc7608576)

[Figura 10 - Diagrama de Comunicação 24](#_Toc7608577)

[Figura 11 - Diagrama de Implantação 25](#_Toc7608578)

**LISTA DE QUADROS**

[Quadro 1 - Requisitos Funcionais do Sistema Joga Junto 16](#_Toc7608591)

[Quadro 2 - Regras de Negócio do aplicativo Joga Junto 17](#_Toc7608592)

[Quadro 3 - Requisitos Não Funcionais do Aplicativo Joga Junto 18](#_Toc7608593)

[Quadro 4 - Requisitos Não Funcionais Tecnológicos do Aplicativo Joga Junto 18](#_Toc7608594)

****SUMÁRIO****

1 Introdução 9

1.1 Objetivo Geral 9

1.2 Objetivos Específicos 9

1.3 Metodologias 10

2 TRABALHOS RELACIONADOS 11

2.1 CHEGA + 11

2.2 DONO DA BOLA 11

2.3 PELADEIROS 12

2.4 Considerações Finais 13

3 TECNOLOGIAS UTILIZADAS 14

3.1 APLICATIVO ANDROID 14

3.1.1 Ionic 14

3.2 API Web service 14

3.2.1 PHP 14

3.2.2 Laravel 15

3.2.3 PostgreSQL 15

4 ESTUDO DE CASO 16

4.1 Contextualização 16

4.2 Levantamento de Requisitos 16

4.2.1 Requisitos Funcionais 16

4.2.2 Regras de Negócio 17

4.2.3 Requisitos Não Funcionais 18

4.2.4 Requisitos Não Funcionais Tecnológicos 18

4.3 Diagramas de análise e modelagem do sistema 18

4.3.1 Diagrama de Casos de Uso Geral 18

4.4 Diagramas de Casos de Usos Complexos 20

4.4.1 Diagrama de Classes Conceitual 20

4.4.2 Diagrama de Classes de Implementação 20

5 JOGA JUNTO 22

5.1 Diagramas de projeto 22

5.1.1 Diagrama Entidade Relacionamento 22

5.1.2 Diagrama de Atividades 23

5.1.3 Diagramas de Comunicação de Casos de Usos Complexos 23

5.1.4 Diagrama de Implantação 25

5.2 Aplicação desenvolvida 26

6 CONCLUSÃO 27

REFERÊNCIAS BIBLIOGRÁFICAS 28

# Introdução

O dia-a-dia atarefado oferece poucas oportunidades para manter contato com grupos de pessoas quem tem gostos em comum, fazendo com que muitas vezes se perca a motivação de praticar atividades esportivas em grupos, que dependem que alguém se proponha a organizar os encontros.

A organização do encontro de um grupo é uma atividade árdua, pois consome tempo para entrar em contato com os participantes do encontro, cobrar a confirmação, ratear os custos e ainda haverá a cobrança por uma melhor organização do evento. Para que esse momento de lazer não se torne mais um fator estressante na rotina, a tecnologia vem como uma ferramenta para desonerar o organizar de várias dessas tarefas, e até mesmo dividir esse trabalho em mais de uma pessoa.

A prática de esportes coletivos é recomendada por especialistas da área de saúde, pois além de contribuir para a saúde física, é muito importante como uma ferramenta para relaxar a mente com algo prazeroso. O software apresentado neste trabalho acadêmico tem como objetivo ser facilitador da organização de práticas de esportes coletivos, auxiliando o organizador a manter contato com os membros, disponibilizar listagem de participantes confirmados para todos os envolvidos, diminuindo a carga de informações solicitadas, ajudando no rateio de valores, e também na interação pós-evento dos membros.

## Objetivo Geral

Criar um produto de software no formato de um aplicativo para Android que ajude as pessoas a organizarem eventos esportivos de maneira simples.

## Objetivos Específicos

Fornecer uma ferramenta para que o usuário possa buscar em toda sua lista de contatos, bem como em locais próximos de sua localização, pessoas para praticar esportes coletivos em uma data e local definidos.

Oferecer uma forma de pessoas que possuem poucos contatos, ou que simplesmente queiram praticar uma atividade esportiva, mas não possuem grupos para tal naquele momento, possam buscar em sua região partidas agendadas que estejam abertas a receber novos participantes.

## Metodologias

Para o desenvolvimento do aplicativo Joga Junto, será escrita uma API (*Application Programming Interface*) na linguagem PHP versão 7, utilizando o framework Laravel na versão 5.7. Os dados serão armazenados em um SGBD (Sistema gerenciador de banco de dados) PostgreSQL versão 11, e para o aplicativo utilizado Android nativo, com a ferramenta Android Studio.

A documentação será desenvolvida utilizando a linguagem visual UML, com a ferramenta Astah (versão estudante), e o banco de dados será modelado em Mysql Workbench.

A escolha por criar uma API para concentrar o processamento e armazenamento dos dados, tem o objetivo de diminuir o uso de recursos dos dispositivos móveis, e a futura disponibilização dos dados em outras plataformas. Além disso, há a necessidade de compartilhamento de informações entre os usuários, alertas de notificações e histórico de avaliações.

Bancos de dados relacionais têm a função de organizar e manter as informações de usuários, grupos qual pertencem. Devido à estimativa de volume de dados, optou-se pelo uso do PostgreSQL, por ser reconhecidamente uma ferramenta robusta, Open source, e a linguagem plpgsql oferece inúmeros recursos para criação de funções e *procedures* que ajudarão a garantir a qualidade do produto. Além disso, conta com uma comunidade muito atuante, que proporciona fácil acesso a soluções para quaisquer desafios encontrados no desenvolvimento.

O trabalho do pré-processamento das informações será executado em ambiente web, utilizando o framework Laravel, com a linguagem PHP. A opção por esta, entre várias alternativas, se dá por ser uma linguagem de fácil acesso, vasto suporte, que permite alto escalonamento da aplicação. O Laravel é um framework que dá suporte a uma série de funções de base, como um ORM (*Object Relation Mapping*), envio de e-mail, controle de autenticação, segurança de nível acesso, e tudo isso com uma estrutura MVC (*Model View Controlller*) muito bem definida, que faz bom uso dos principais padrões de projeto.

# TRABALHOS RELACIONADOS

## CHEGA +

O Chega+ é o aplicativo para organizar grupos de futebol mais bem avaliado entre os encontrados. Permite que todos os jogadores do grupo participem da administração das partidas. Conta com recursos como confirmação de presença, sorteio de times, atribuição de notas aos jogadores, controle financeiro entre outros.

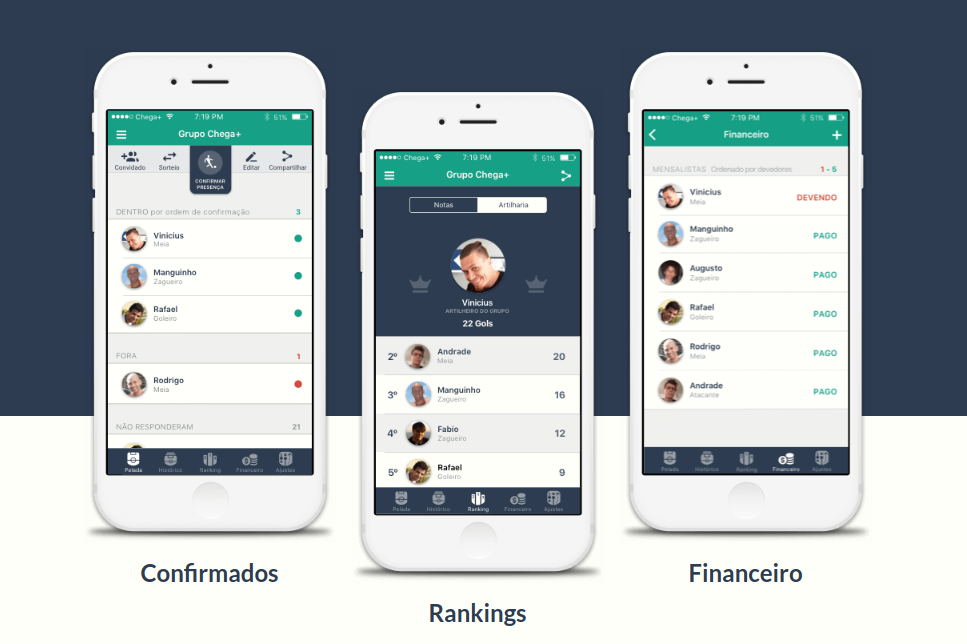


Figura - APLICATIVO CHEGA +

## DONO DA BOLA

Dono da bola é um aplicativo facilitador para o organizador da pelada, oferece funções como inserir contatos da lista, despesas, controle de recebimento, e ainda um recurso interessante de envio de e-mail com informações sobre última pelada jogada para os demais jogadores. O aplicativo é disponibilizado apenas na plataforma IOS (APPLESTORE, 2018).



Figura - APLICATIVO DONO DA BOLA

## PELADEIROS

Peladeiros, assim como os demais, foca na gestão do evento esportivo. O que o difere dos demais são os recursos disponíveis, pois conta com funções básicas como criação de times, controle financeiro, artilharia, histórico e outros recursos mais avançados como distribuição de jogadores por nível técnico para equilíbrio das equipes, criação de grupos, envio de SMS para a lista dos envolvidos com avisos relacionados ao evento. Um ponto importante é que não conta com lista de confirmação de presentes.

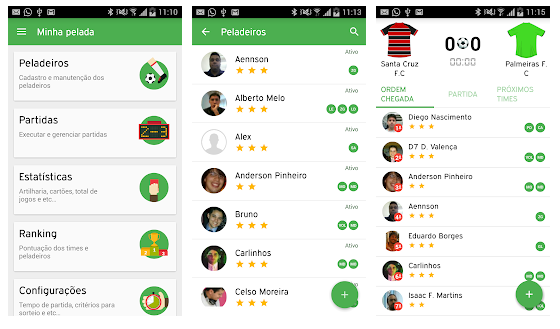


Figura - aplicativo peladeiros i

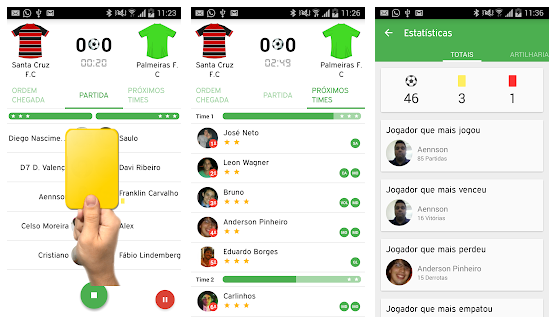


Figura - aplicativo peladeiros ii

## Considerações Finais

Analisando as soluções que tem propósito semelhante ao projeto, alguns pontos diferem, como no caso do Peladeiros, que possui uma série de funcionalidades na modalidade gratuita, mas não possui confirmação, e este é um recurso que em grupos pequenos tem grande importância.

No caso do Aplicativo Dono da Bola, a facilidade e simplicidade se destacam, bem como a clareza das informações apresentadas, no entanto atende apenas o público que dispõe de dispositivos com sistema IOS.

Já o Aplicativo Chega+ possui recursos necessários para organização de partidas entre amigos, o ponto baixo desse aplicativo é a execução, uma vez que possui muitas reclamações sobre funções simples, como sorteio de equipes.

O aplicativo proposto irá possibilitar a interação entre jogadores e organizadores que não se conhecem, não possuem contatos para encontrar eventos e jogadores disponíveis na região, função que os aplicativos analisados não apresentam.

# TECNOLOGIAS UTILIZADAS

## APLICATIVO ANDROID

### Ionic

Framework de desenvolvimento de aplicações híbridas, onde com apenas um código fonte, pode-se criar aplicações para Android, IOS, Windows Phone e até mesmo para web.

Não é necessário possuir maquinas virtuais para testes, uma vez que a aplicação pode ser totalmente desenvolvida no browser, o que consome menos recursos que as aplicações nativas que requerem emuladores para desenvolver.

Atualmente está na quarta versão, em suas primeiras versões utilizava o AngularJS com os recursos nos navegadores.

A partir da versão 2.0, quando a Google lançou o novo Angular 2.0 que utiliza o TypeScript, que é um superset do Javascript, onde se desenvolve orientado a objetos, e então o código é compilado para javascript para ser executado.

Ionic se baseia em webcomponent, utilizando javascript, css e outros recursos comuns da web. Além disso, da versão 4 em diante, o Ionic permite que se desenvolva utilizando outros frameworks e bibliotecas como VueJS, React, jQuery etc, e portanto é enquadrado como um PWA, Progressive Web Application (CITAÇÃO, 2019).

## API Web service

### PHP

Por razão de uma comunidade atuante e do fácil acesso a bases de conhecimento, o PHP é uma das linguagens mais populares quando o assunto é programação para a web. Desde seus primórdios, foi pensada para o uso rápido e prático. Por causa dessa filosofia, a sua fama passou a ser de uma linguagem com baixo nível de segurança e com muitos *bugs*. Com o passar do tempo a linguagem evoluiu e absorveu os paradigmas da programação orientada a objetos, e tornou-se uma robusta ferramenta para o desenvolvimento não só de *websites*, mas também de avançados sistemas dos mais variados temas.

A escolha pela linguagem se deve pela experiência que a equipe possui e pela vasta gama de ferramentas que a linguagem proporciona, de modo a facilitar o desenvolvimento da API.

### Laravel

Em uma enorme gama de opções de frameworks PHP existentes, o Laravel se destaca como uma ótima opção para desenvolvimento de aplicações. Possui uma filosofia de “código elegante”, é favorável para um desenvolvimento organizado e de fácil entendimento. Possui um ORM prático e poderoso que agiliza o desenvolvimento das entidades e seus relacionamentos mais complexos, e um roteamento de fácil configuração para implementação da API, além de um seguro módulo de autenticação que é extremamente necessário para o foco que a aplicação toma.

A escolha do framework se deve pela experiência que a equipe possui, e pela grande quantidade de ferramentas que já existem para o framework, visando facilitar o desenvolvimento, como Ferramentas de Mailer, Routing, ORM entre outras.

### PostgreSQL

O PostgreSQL tem se destacado no mercado como um dos melhores e mais robustos sistemas gerenciadores de bancos de dados relacional, possuindo recursos muito importantes como controle de concorrência multi versão, que evita bloqueios de registros conhecidos como “*lock”,* isolamento de transações que evita que um usuário faça leitura de dados de outro usuário ainda não *commitados.* Além disso, possui licença BSD (*Berkeley Software Distribution,* considerado como de domínio público e pode ser modificado sem nenhuma restrição).

# ESTUDO DE CASO

## Contextualização

Encontrar um grupo para prática de esportes coletivos é uma tarefa complicada, pois demanda tempo, muitos contatos, e algumas vezes a falta de retorno de colegas acaba desmotivando, tirando o interesse e contribuindo para uma vida sedentária.

Observando esse nicho, a possibilidade de intervir e facilitar o contato, organização e realização de eventos, surgiu a proposta do projeto do aplicativo Joga Junto.

## Levantamento de Requisitos

### Requisitos Funcionais

Quadro - Requisitos Funcionais do Sistema Joga Junto

|  |  |
| --- | --- |
| **Código** | **Requisito Funcional** |
| RF 01 | O sistema deve gerenciar o acesso dos usuários ao sistema por meio de *login* e senha, rede social ou conta Google. |
| RF 02 | O sistema deve manter os usuários com os seguintes dados: *login*, senha, nome, e-mail, telefone e endereço, esportes praticados, posições que atua em cada esporte. |
| RF 03 | O sistema deve manter os locais para praticas esportivas com os seguintes dados: nome, descrição, endereço, contato, esportes disponíveis, horário de atendimento e usuário responsável. |
| RF 04 | O sistema deve manter eventos esportivos com os seguintes dados: local, usuário responsável pelo evento, data, horário, descrição, esporte, quantidade de vagas, abertura ao público, valor de custo. |
| RF 05 | O sistema deve permitir que o organizador de um evento envie convites para seus contatos, por meio de link, e-mail ou redes sociais. |
| RF 06 | O sistema deve permitir que os usuários convidados possam aceitar ou declinar a participação em eventos quais foram convidados, ou ainda cancelar confirmações já efetuadas. |
| RF 07 | O Sistema deve permitir que o usuário responsável possa abrir evento para participação de terceiros, e aceitar ou declinar solicitações para entrar no seu evento. |
| RF 08 | O Sistema deve permitir que os usuários possam avaliar a participação dos colegas com uma nota e um comentário curto, e também o local esportivo. |
| RF 09 | O Sistema deve permitir que os usuários responsáveis por locais realizem a confirmação de eventos, e também o cancelamento do mesmo informando os organizadores de eventos agendados. |
| RF 10 | O Sistema deve permitir que o usuário responsável pelo evento possa realizar o cancelamento do mesmo, informando os usuários participantes confirmados, candidatos ou pendentes de confirmação. |
| RF 11 | O Sistema deve permitir que os usuários possam realizar o cancelamento de sua participação em um evento já confirmado, notificando o usuário organizador do evento |
| RF 12 | O Sistema deve permitir que os usuários, ao criar um evento esportivo, se não encontrar o local, possa cadastrar o mesmo, informando campos: nome, endereço, referencia, como chegar. O local esportivo ficará disponível para o seu evento, e estará disponível para os demais usuários após moderação da administração. |
| RF 13 | O Sistema deve permitir que os usuários possam requisitar a propriedade de um local previamente cadastrado e publicado, enviando documentos que comprovem a relação do usuário com o local. |
| RF 14 | O Sistema deve permitir que os usuários possam fazer denúncias descritivas de outros usuários, ou locais esportivos que por ventura, venham violar alguma lei vigente no pais. |

Fonte: Autoria própria (2019)

### Regras de Negócio

Com os requisitos expostos, alguns pontos importantes precisam ser verificados, para que o aplicativo possua um bom funcionamento.

Quadro - Regras de Negócio do aplicativo Joga Junto

|  |  |  |
| --- | --- | --- |
| **Código** | **RF** | **Regra de Negócio** |
| RN 1 | RF 01 | Para acessar qualquer área do sistema, o usuário deve estar logado |
| RN 2 | RF 02 | Para cadastrar-se no sistema, os usuários deverão ter no mínimo 18 anos de idade. |
| RN 3 | RF 03 | Toda partida precisa ter um local esportivo previamente cadastrado. |
| RN 4 | RF 03 | Um Local esportivo precisa ter um endereço validado por um CEP |
| RN 5 | RF 04 | Criação de eventos somente em data futura. |
| RN 6 | RF 04 | Eventos esportivos deverão ter um limite máximo de participantes definido pelo organizador, respeitando os limites mínimo e máximo previamente cadastrado no tipo de esporte. |
| RN 7 | RF 05 RF 06 | Eventos esportivos visíveis somente os abertos ao publico, e que ainda não tenham confirmados o numero máximo de participantes. |
| RN 8 | RF 06 | Buscas de eventos esportivos por localidade serão limitados a um raio de 20km |
| RN 9 | RF 05 | Com um evento esportivo criado, o organizador pode enviar no máximo o número limite do esporte em convites a cada 15 minutos. |
| RN 10 | RF 06 RF 11 | O cancelamento de participação em eventos onde o usuário já confirmou sua presença deve seguir com uma justificativa, que será encaminhada ao organizador |
| RN 11 | RF 04  RF 11 | A exclusão de um usuário confirmado em um evento por parte do organizador, deve seguir com uma justificativa, que será encaminhada ao usuário. |
| RN 12 | RF 04 | O cancelamento de um evento por parte do organizador, deve seguir com uma justificativa, que será enviada a todos os envolvidos. |
| RN 13 | RF 08 | Após a realização do evento, os usuários pode fazer suas avaliações do local, e de outros membros em no máximo 24 horas. |

Fonte: Autoria própria (2019)

### Requisitos Não Funcionais

Quadro - Requisitos Não Funcionais do Aplicativo Joga Junto

|  |  |
| --- | --- |
| **Código** | **Requisito Não Funcional Tecnológico** |
| RNF 01 | O sistema deve usar a API do Google busca partidas disponíveis próximas da região do usuário |
| RNF 02 | O sistema deve usar a API viacep busca de CEP. |

Fonte: Autoria própria (2019)

### Requisitos Não Funcionais Tecnológicos

Quadro - Requisitos Não Funcionais Tecnológicos do Aplicativo Joga Junto

|  |  |
| --- | --- |
| **Código** | **Requisito Não Funcional Tecnológico** |
| RNFT 01 | O sistema deve dispor de uma API escrita em PHP 7.2, utilizando o framework Laravel 5.7. |
| RNFT 02 | O sistema deve manter os dados em banco de dados relacional PostgreSQL versão 11. |
| RNFT 03 | O sistema deve se comunicar com a API por meio do protocolo HTTP, utilizando padrão RESTFul |
| RNFT 04 | O Sistema deve ser escrito em linguagem IONIC 3, utilizando Angular |

Fonte: Autoria própria (2019)

## Diagramas de análise e modelagem do sistema

### Diagrama de Casos de Uso Geral

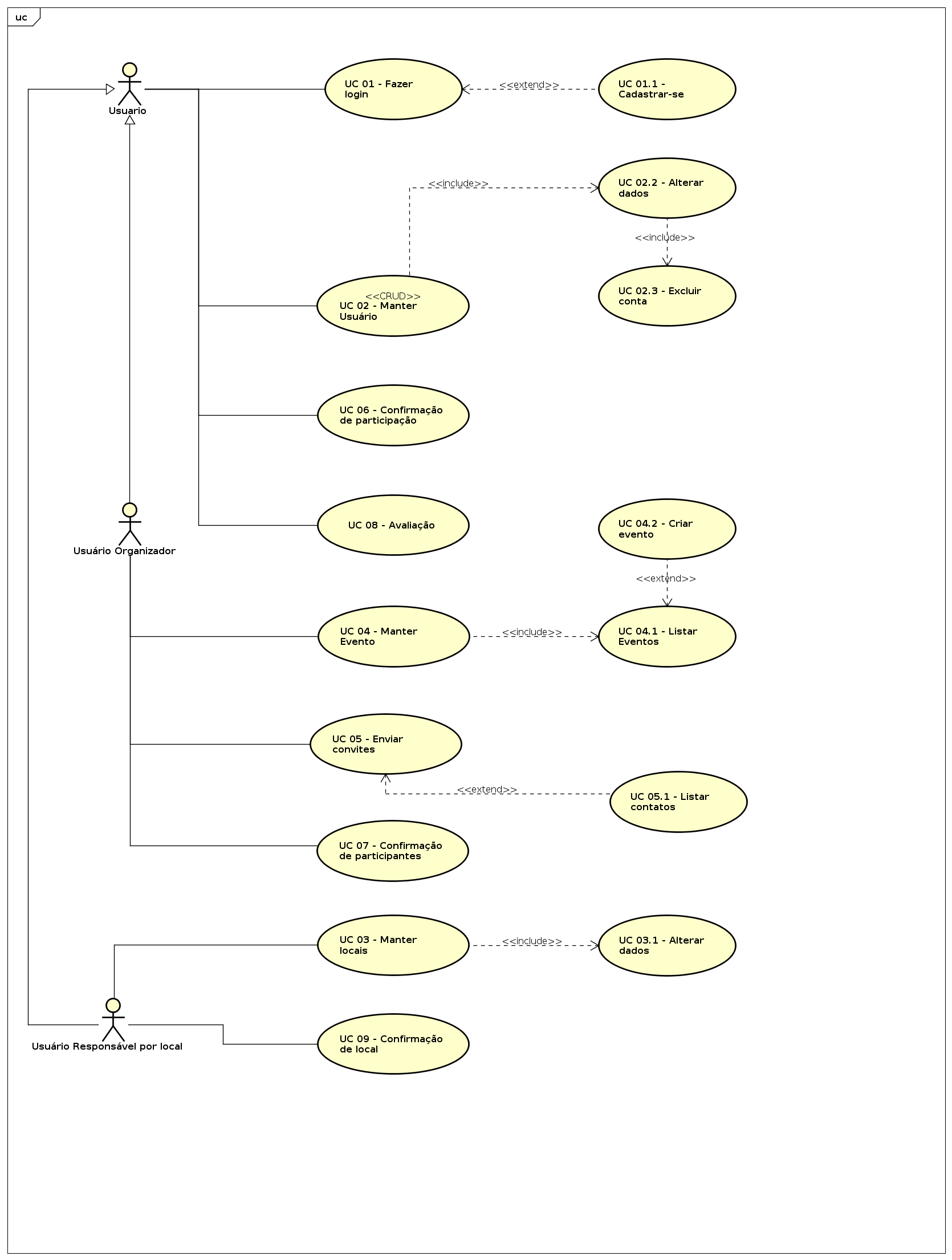


Figura - Diagrama de Casos de Uso

## Diagramas de Casos de Usos Complexos

### Diagrama de Classes Conceitual

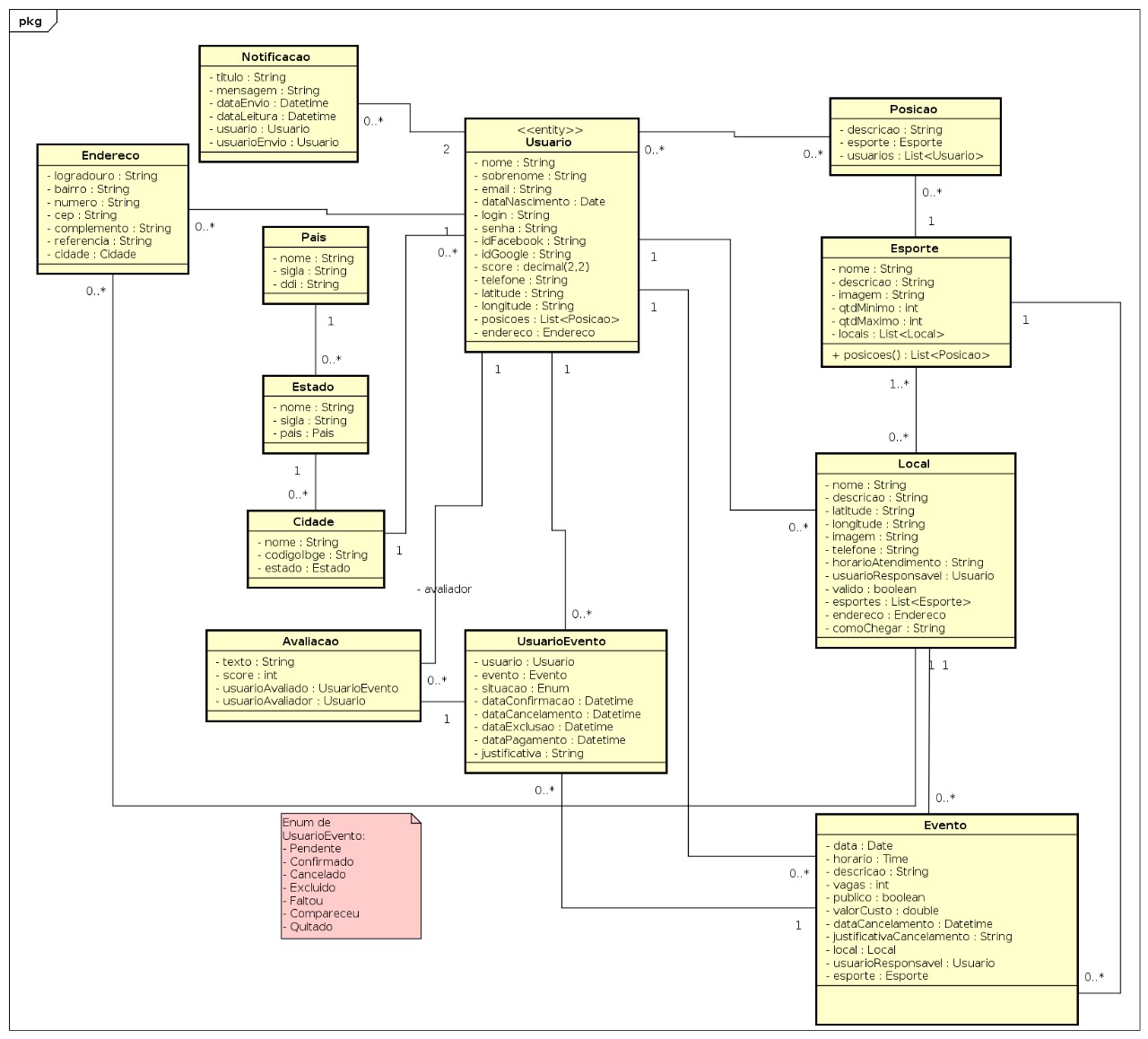


Figura - Diagrama de Classes Conceitual

### Diagrama de Classes de Implementação

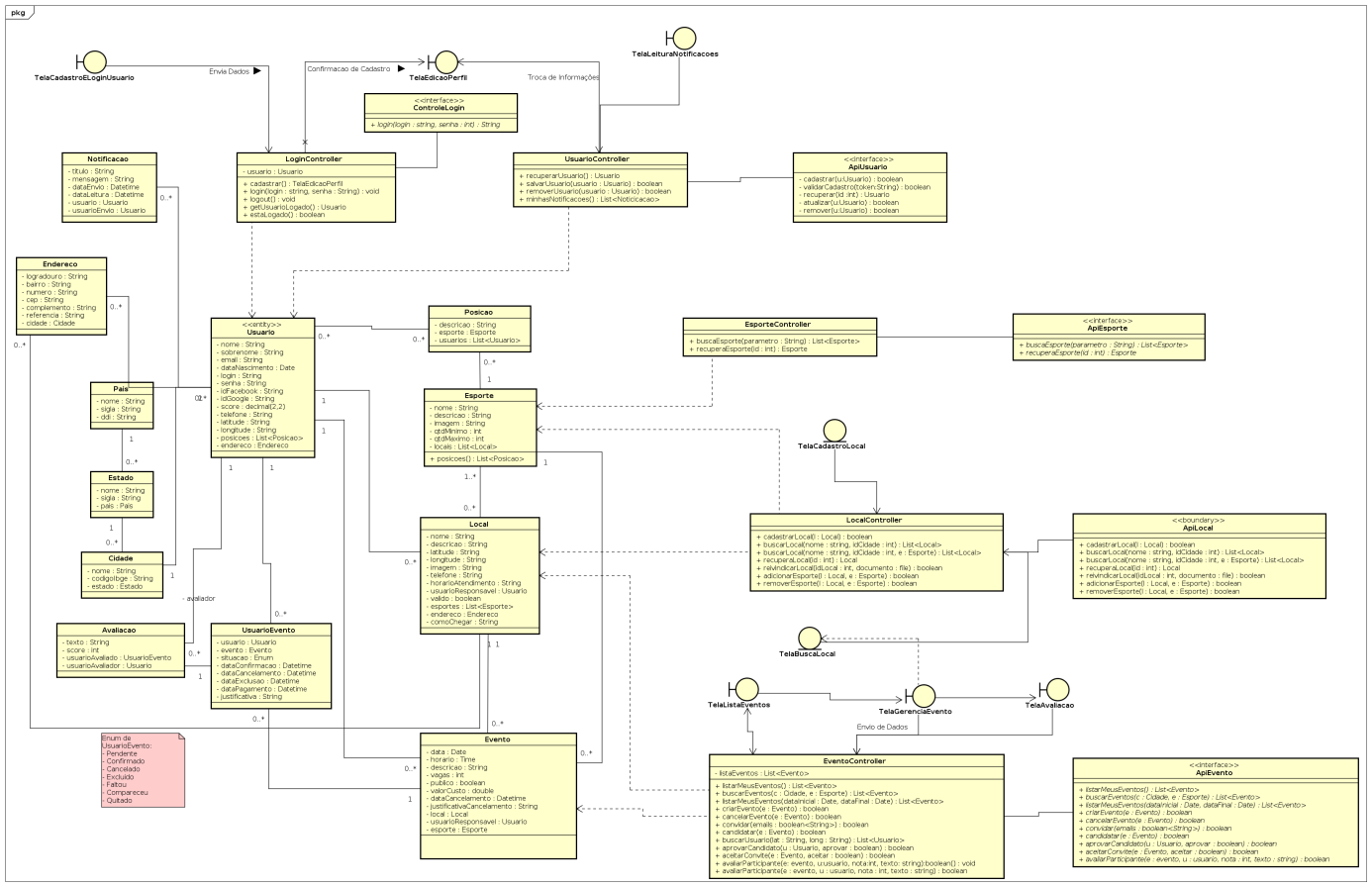


Figura - Diagrama de Classes de Implementação

# JOGA JUNTO

## Diagramas de projeto

### Diagrama Entidade Relacionamento

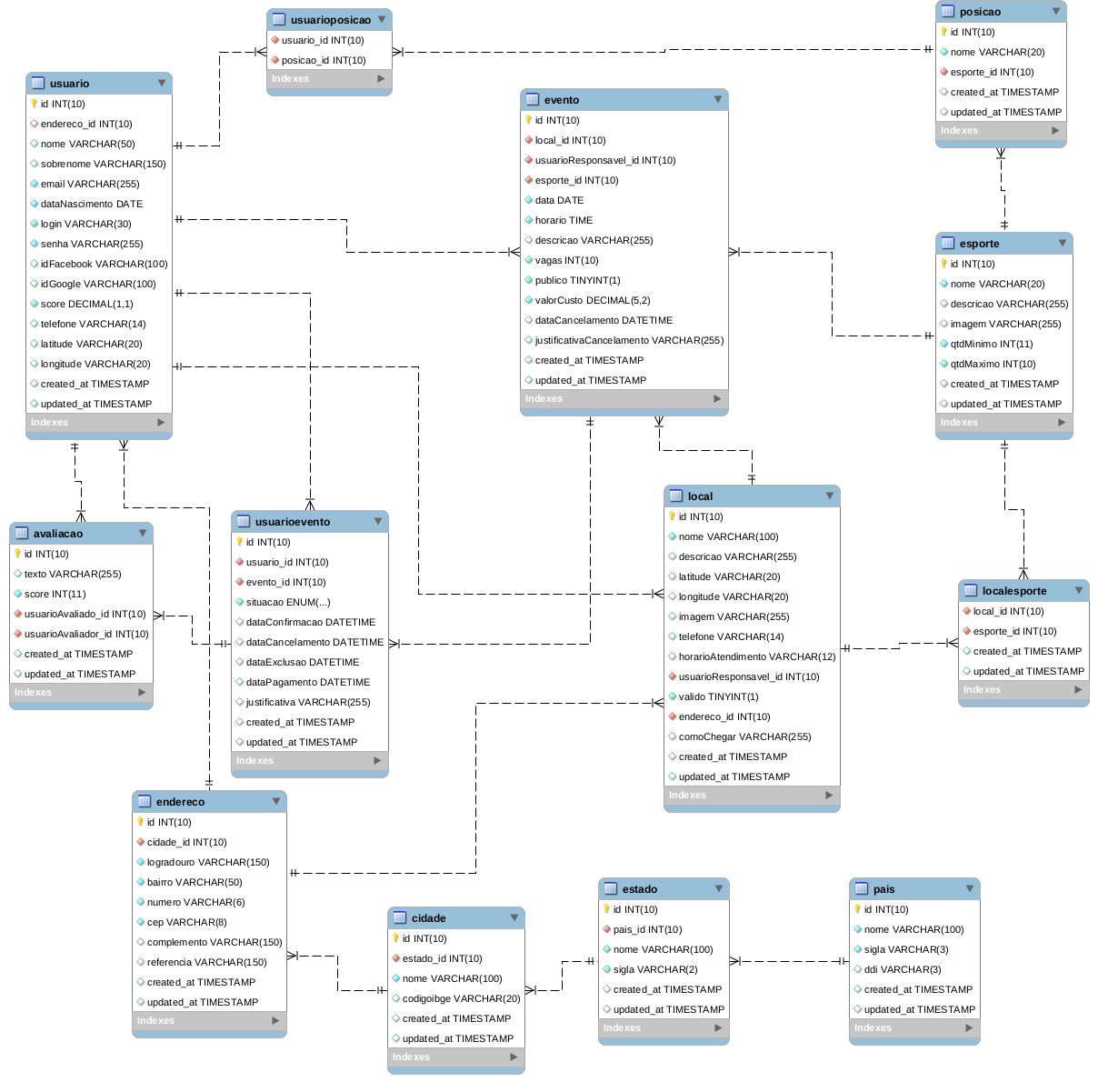


Figura - Diagrama de Entidade-Relacionamento

### Diagrama de Atividades

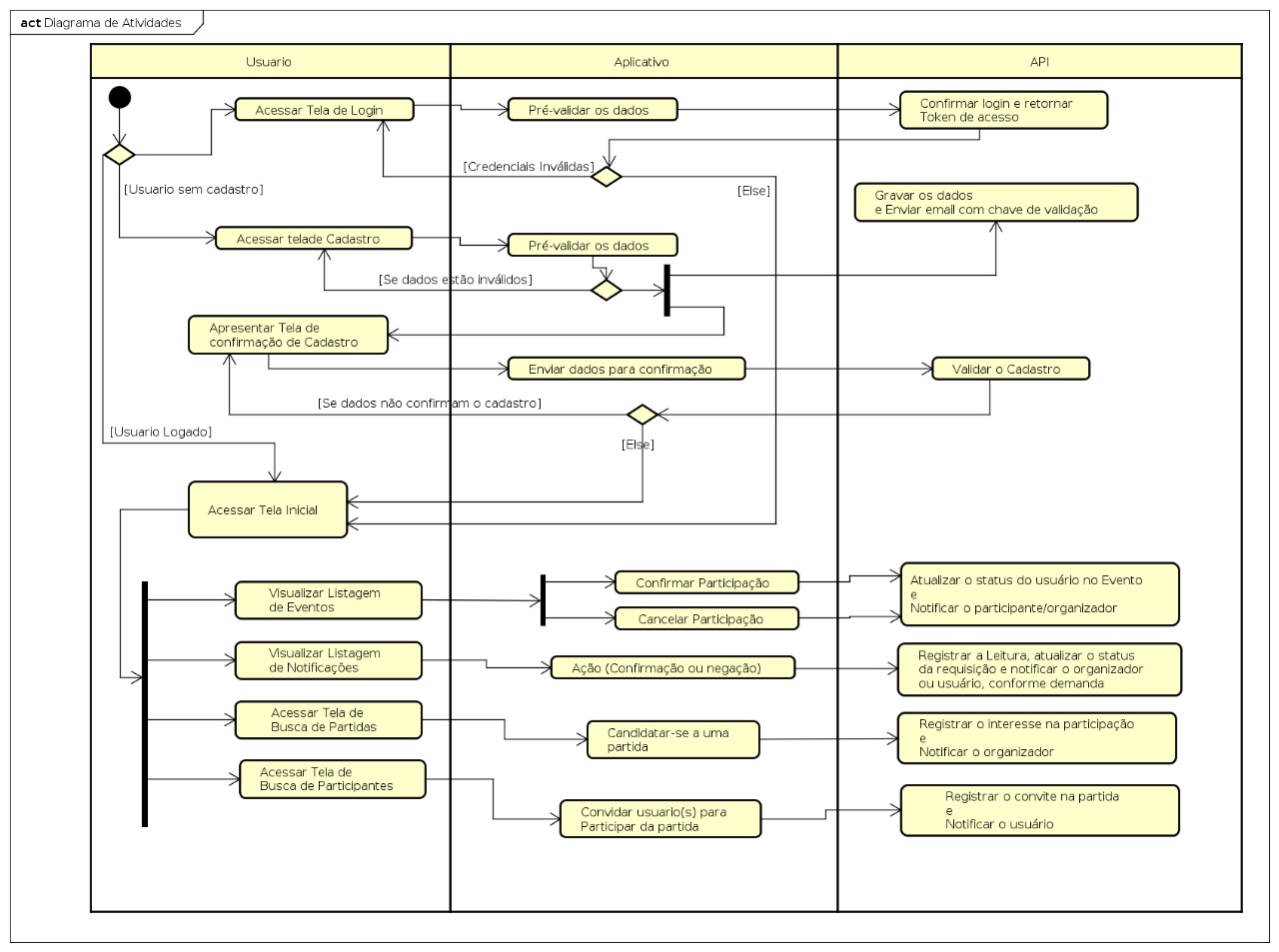


Figura - Diagrama de Atividades

### Diagramas de Comunicação de Casos de Usos Complexos

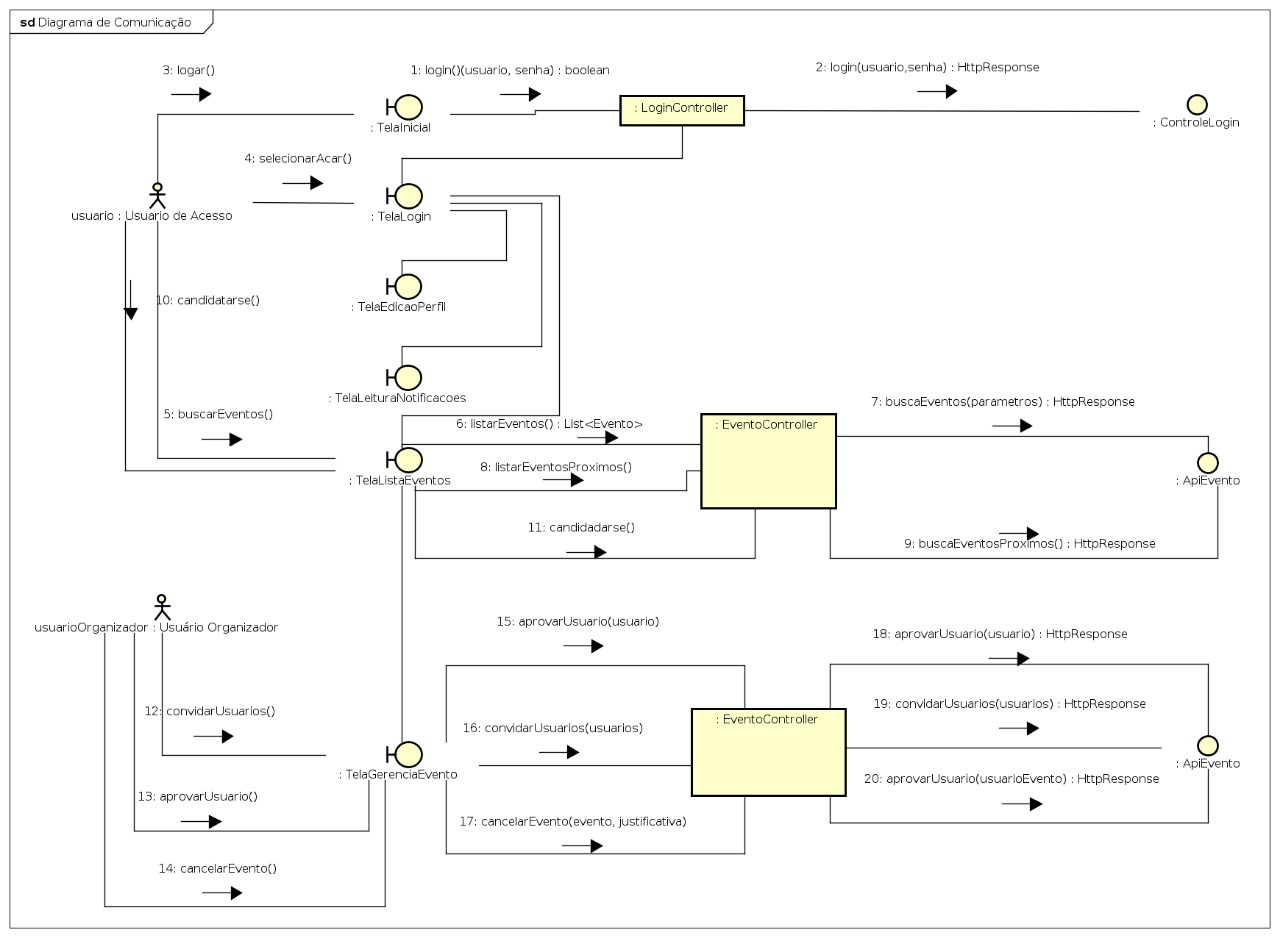


Figura - Diagrama de Comunicação

### Diagrama de Implantação

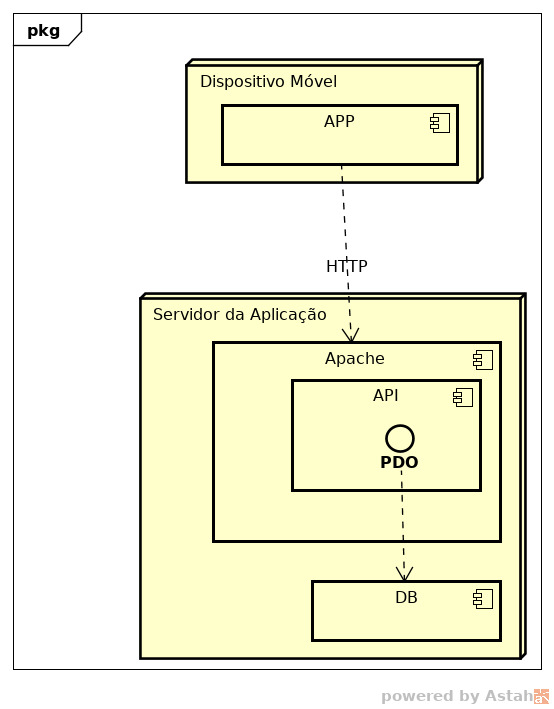


Figura - Diagrama de Implantação

## Aplicação desenvolvida

Descrição do funcionamento do sistema com captura de telas e eventualmente com explicação de códigos complexos.

# CONCLUSÃO

# REFERÊNCIAS BIBLIOGRÁFICAS

Em ordem alfabética!!!