

Universidade do Estado do Rio de Janeiro
Instituto de Matemática e Estatística
Departamento de Informática e Ciência da Computação
Disciplina: Análise e Projeto de Sistemas

Extensão do Airbnb: *SafeServices*

Grupo: Camila Gusmão
Gabriel Cardoso
Hugo Lima
Murilo de Jesus
Victoria Velasco

Rio de Janeiro
Agosto/2015

Sumário

[Capítulo 1 - Introdução](#)

[Capítulo 2 - Diagrama de Casos de Uso](#)

[Capítulo 3 - Diagrama de Classes do Domínio](#)

[Capítulo 4 - Diagrama de Sequência do Sistema](#)

[Capítulo 5 - Diagrama de Pacotes](#)

[Capítulo 6 - Diagrama de Classes do Projeto](#)

[Capítulo 7 - Diagrama de Sequência do Projeto](#)

[Capítulo 8 - Diagrama de Atividade](#)

[Capítulo 9 - Diagrama de Estados](#)

Lista de Figuras

[Figura 1- Diagrama de Contexto](#)

[Figura 2- Diagrama de Casos de Uso](#)

[Figura 3- Diagrama de Classes do Domínio](#)

[Figura 4- Diagrama de Sequencia](#)

[Figura 5- Diagrama de Sequencia Pagamento Cartao de Crédito](#)

[Figura 6- Diagrama de Sequencia Pagamento PayPal](#)

[Figura 7- Diagrama de Sequencia Pagamento Boleto Bancário](#)

[Figura 8- Diagrama de Pacotes](#)

[Figura 9- Diagrama de Classes do Projeto](#)

[Figura 10- Diagrama de Sequência do Projeto](#)

[Figura 11- Diagrama de Atividade](#)

[Figura 12- Diagrama de Estados](#)

Resumo

Este trabalho provê a comunicação de pessoas em diferentes lugares do mundo, afim de alugar uma acomodação e contratar serviços quando forem necessários, utilizando o aplicativo Airbnb com a implementação de uma extensão. Nesta extensão, o usuário terá diversas opções de serviços confiáveis na região em que estará localizada sua acomodação.

Para o desenvolvimento do trabalho foi utilizada a UML como ferramenta para modelagem de sistemas. A Linguagem de Modelagem Unificada (UML) é uma linguagem visual para especificar, construir e documentar os artefatos dos sistemas [1]. Além disso, ela ajuda a compreender melhor o sistema que será desenvolvido.

Capítulo 1– Introdução

Muitas pessoas, quando desejam viajar para outros países, enfrentam dificuldades em encontrar uma acomodação. Neste capítulo, apresentaremos o aplicativo Airbnb com a extensão SafeServices, que tem o propósito de facilitar o dia-a-dia das pessoas nos países e regiões em que se localizam.

1.1. **Airbnb**

O aplicativo Airbnb propõe uma nova forma de viajar, que se baseia na confiança entre os usuários. Seu objetivo principal é facilitar a comunicação entre pessoas de diversos países, ajudando quem deseja alugar um imóvel por tempo determinado ou quem deseje viajar otimizando seus gastos com acomodações. Deste modo, os anfitriões cadastram seus imóveis com suas respectivas informações e fotos. O usuário escolhe a acomodação na região que lhe interessa, analisa as informações, assim como os comentários e indicações de hóspedes anteriores, e entra em contato com o anfitrião, caso queira alugar a acomodação.

1.2. **SafeServices**

Com extensão para o aplicativo Airbnb, apresentamos o SafeServices, que será uma funcionalidade de guia de serviços, que juntando com os dados anterior a extensão, o usuário poderá escolher um serviço na região/país do seu interesse. Cada serviço é cadastrado por um anfitrião, que ao preencher as informações sobre seu imóvel, preenche também informações(nome, endereço, telefone) sobre um profissional de sua confiança que desempenha determinado serviço, como: empregada, mecânico, cuidador de idoso, babá, médico, enfermeira, advogado e etc.

O SafeService permite que o usuário contrate serviços a qualquer momento de sua estadia, tendo a facilidade de encontrar um serviço de confiança, perto de sua acomodação, sem precisar perder tempo procurando profissionais pela região em que está, o que seria difícil sem a extensão, pois os serviços oferecidos podem não ser confiáveis e a comunicação entre as pessoas também seria algo que dificultaria. Assim, o usuário pode selecionar o serviço desejado, escolher o profissional e contrata-lo. Ao escolher o profissional que

deseja, suas informações aparecerão para o usuário, que poderá ligar, mandar mensagem ou e-mail(para aqueles que possuem), acertando os detalhes(data, local, pagamento).

O pagamento destes serviços pode ser feito de duas maneiras: através da conta do profissional ou pessoalmente em dinheiro. A forma de pagamento deverá ser acertada entre o usuário e o profissional, sendo a melhor forma para ambos.

A medida que o usuário utiliza os serviços da extensão, o mesmo poderá avalia-los, caso queira, dando um “feedback” para que outras pessoas possam ver se o profissional é realmente confiável e desempenha um bom trabalho ou não. Desta forma, a extensão segue a mesma base do aplicativo principal, de trabalhar com a honestidade e confiança das pessoas.

1.3. Diagrama de Contexto

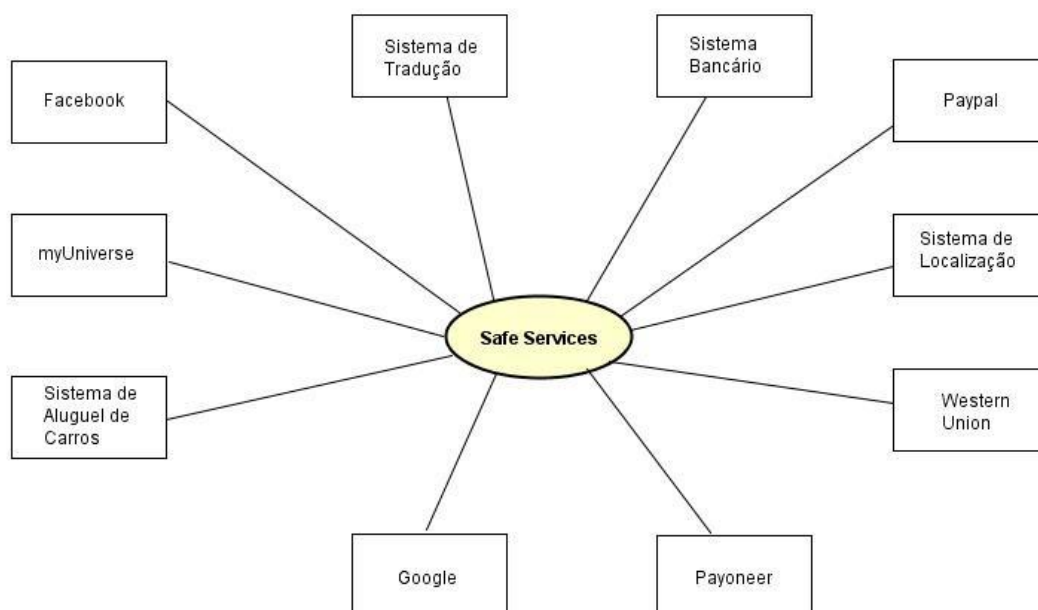


Figura 1- Diagrama de Contexto

Este diagrama apresenta oito sistemas externos ao sistema que está sendo desenvolvido:

1. Sistema Bancário

Presente no *Airbnb* e incorporado aos sistemas externos da extensão *SafeServices*, seu objetivo é facilitar as transações financeiras entre anfitriões, prestadores de serviços e futuros hóspedes, para pagamento dos serviços disponibilizados no site.

2. Sistema de Localização

Desenvolvido para a própria extensão, o Sistema de Localização, utilizará as informações obtidas através do GPS, com autorização do usuário, para fornecer informações sobre serviços cadastrados no site e próximos ao local que se encontra o usuário.

3. Sistema de Aluguel de Carros

O sistema apresenta opções de carros para aluguel de acordo com a empresa indicada pelo anfitrião. Este sistema já existe e será importado para o *SafeServices*.

4. MyUniverse

O sistema tem como objetivo auxiliar os hóspedes a terem informações sobre o local de suas acomodações, como telefones de emergência, pontos turísticos, e previsão do tempo. Este sistema é uma extensão já implementada para o *Airbnb*.

5. Google

Sistema utilizado para auxílio e coleta de informações de usuários que já possuem cadastro no *Google* e desejam utilizar as informações pessoais vinculadas à essa conta, para realizar seu cadastro no *Airbnb*. Este sistema já presente no *Airbnb*, será incorporado à extensão *SafeServices*.

6. Facebook

Sistema utilizado para auxílio e coleta de informações de usuários que já possuem cadastro na rede social *Facebook* e desejam utilizar as informações pessoais vinculadas à essa conta, para realizar seu cadastro no *Airbnb*. Este sistema já presente no *Airbnb*, será incorporado à extensão *SafeServices*.

7. Paypal

Sistema de pagamento online, já vinculado ao *Airbnb*. O *Paypal* é um sistema que permite a transferência de dinheiro entre indivíduos ou negociantes usando um endereço de e-mail, assim, evitando métodos tradicionais como cheques e boleto bancário.

8. Payoneer

Sistema de pagamento online, já vinculado ao *Airbnb*. O *Payoneer* é um sistema que permite a transferência de dinheiro entre indivíduos ou negociantes em escala global, com taxas inferiores ao *Paypal*, mas menos presente no mercado que o mesmo.

9. Western Union

Sistema de pagamento online, já vinculado ao *Airbnb*. O *Western Union* funciona como uma agência bancária virtual e segue o ideal do *Paypal* e *Payoneer*.

10. ACH

Sistema de pagamento online, já vinculado ao *Airbnb*. O *Automated Clearing House (ACH)* é uma rede virtual para transações financeiras no Estados Unidos, inclui atividades como depósito direto, boletos, entre outros.

Capítulo 2 - Diagrama de Casos de Uso

O diagrama de casos de uso é um diagrama da UML que documenta o que o sistema do ponto de vista do usuário, analisando as principais funcionalidades do sistema e facilitando a sua comunicação com o analista de sistemas. Com esse diagrama, pode-se fazer o levantamento e validação dos requisitos funcionais do sistema. Este diagrama é composto por:

- * Atores: são entidades externas com quem o sistema interage, ou seja, um tipo de usuário do sistema. Um ator representa um papel no sistema, mas um papel pode ser representado por diversos atores;
- * Caso de uso: Representa uma funcionalidade do sistema desempenhada por um ator ou por outro caso de uso.

O conjunto de casos de uso pode ser dividido em dois tipos:

- Primário: que representam os objetivos básicos dos atores ao acessar o sistema;
 - Secundários: que não trazem benefícios diretos para os atores, mas são necessários para que o sistema funcione adequadamente.
- * Relacionamentos: É a interação entre os atores e o conjunto de casos de uso.

Os relacionamentos podem ser do tipo:

- Associativo entre atores e o conjunto de casos de uso;
- Generalização entre os atores;
- Dependência: extends e includes.

2.1. Diagrama de Casos de Uso do Sistema

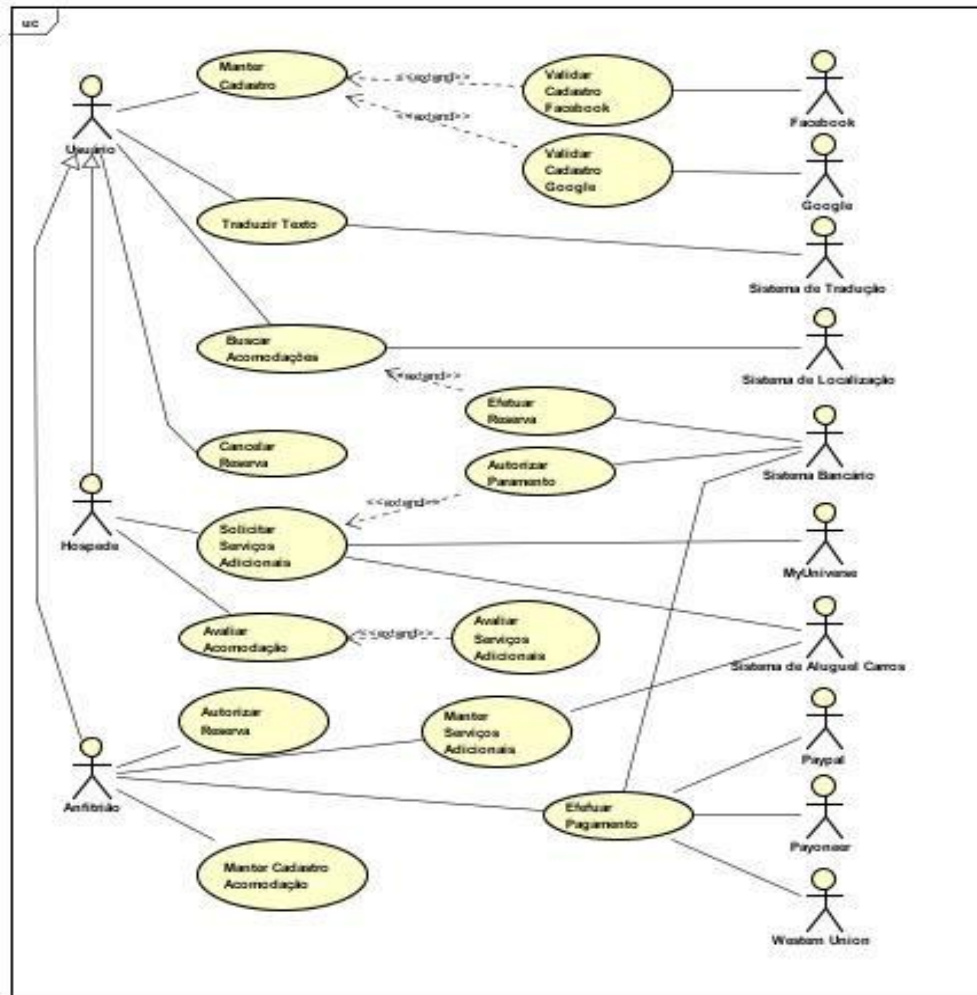


Figura 2- Diagrama de Casos de Uso

2.2. Descrição do Diagrama Casos de Uso

2.2.1. Descrição dos Atores

Os atores são representados por um boneco com o nome do ator. Neste caso temos como atores:

Usuário:

O Usuário é uma pessoa que irá viajar e busca por uma acomodação na região/cidade de destino.

Anfitrião:

O Anfitrião é um usuário que cadastra seu imóvel para locação e cadastra serviços adicionais.

Hospede:

O Hospede é um usuário que solicita serviços adicionais na região que esta acomodada e os avalia.

Facebook:

O Facebook é uma rede social onde pessoas fazem seu perfil online, publicam fotos, vídeos, textos, status e se conectam com seus amigos para ver suas publicações.

Google:

O Google é uma rede online de pesquisa que contém diversas extensões, possibilitando que seu usuário possa editar textos, fazer tabelas, apresentações e ter um perfil social para publicar fotos, vídeos, textos e seguir seus amigos.

Sistema de Tradução:

O Sistema de Tradução é um sistema que permite ao usuário traduzir textos para sua língua materna.

Paypal:

O Paypal é um sistema que permite a transferência de dinheiro entre indivíduos ou negociantes, usando um endereço de e-mail.

Sistema Bancário:

O Sistema Bancário é um sistema que permite a transferência de dinheiro entre indivíduos ou negociantes, usando contas bancárias.

Payoneer:

O Payoneer é um sistema que permite o usuário comprar um cartão pré-pago com créditos para fazer pagamentos.

Western Union:

O Western Union é um sistema que permite a transferência de dinheiro de forma rápida.

Sistema Localização:

O Sistema de Localização é um sistema que permite ao usuário saber a sua localização no momento em que deseja, através de satélites.

Sistema de Aluguel de Carros:

O Sistema de Aluguel de Carros é um sistema que mostra os carros disponíveis para aluguel na região em que o usuário deseja.

MyUniverse:

O MyUniverse é um aplicativo de guia turístico que facilita a comunicação de seus usuários em lugares com idiomas diferentes do seu e além disso mostra ao usuário o que tem na região em que ele se encontra.

2.2.2. Descrição dos Casos de Uso

Um caso de uso define uma funcionalidade do sistema. É representado por eclipses com o nome do caso de uso. Neste caso temos os seguintes casos de uso:

Manter Cadastro:

Neste caso de uso o usuário consegue se cadastrar no Airbnb, via criação de conta com e-mail (ou se logando pelo Facebook ou Google). O sistema faz a validação das informações fornecidas e o cadastro é concluído. Também é permitido ao usuário alterar seus dados cadastrais e cancelar o mesmo.

Validar Cadastro Facebook:

Quando o usuário deseja se cadastrar ao sistema do Airbnb, é possível efetua-lo através da sua conta do Facebook, poupando ao usuário tempo, pois não será necessário preencher os campos com suas informações, estas serão impostadas da sua conta do Facebook.

Validar Cadastro Google:

Ao acessar o sistema para efetuar o cadastro no Airbnb, o usuário tem a opção de realizar isto através da sua conta Google, não perdendo tempo preenchendo os campos com suas informações, pois estas serão importadas do Google pelo sistema.

Traduzir Texto:

O usuário entra no site e caso a pagina não esteja na sua língua materna, o mesmo pode selecionar o idioma que deseja através de uma barra de idiomas que fica no canto inferior direito da tela. Caso o usuário queira alterar o idioma mais de uma vez e em páginas diferentes, o mesmo poderá fazer isso acessando a mesma barra de idiomas que fica disponível em todas as páginas do site.

Buscar por acomodação:

Nome: Buscar por acomodação.

Identificador: CSU01

Importância:

- Risco: alto.
- Prioridade: alta.

Sumário: Cliente usa sistema para encontrar um imóvel para se acomodar.

Ator primário: Cliente

Atores Secundários:

- GPS

Pré-condições:

- O cliente deve possuir cadastro no sistema
- O cliente deve estar logado

Fluxo Principal:

- 1) Hospede escolhe um local para se acomodar, as datas de chegada e partida, a quantidade de hóspedes e outros filtros que lhe interesse.
- 2) Sistema apresenta todas as acomodações que fazem atendam os filtros selecionados.
- 3) Hospede confere as informações fornecidas a partir dos filtros por ele selecionados.
- 4) Hospede seleciona o imóvel de sua preferência.
- 5) Sistema apresenta as informações detalhadas sobre o imóvel que o cliente selecionou.
- 6) Hospede confirma a solicitação de reserva.
- 7) Sistema informa valores de estadia e porcentagem de garantia de reserva.
- 8) Hospede seleciona o modo de pagamento e informa os dados de cartão.
- 9) Sistema notifica solicitação de estadia, envia dados de reserva para anfitrião.
- 10) Anfitrião confirma estadia, sistema confirma pagamento e o caso de uso termina.

Fluxos Alternativos:

- I. Fluxo Alternativo: Hóspede deseja alterar os filtros de busca por acomodação
 - a) Se o hóspede não encontrar uma acomodação de sua preferência que esteja nos filtros selecionados por ele anteriormente, ele tem a opção de refinar os filtros.
 - b) Caso o hóspede refine, ele seleciona a opção 'Mais Filtros' e o caso de uso retorna para o passo 2.
- II. Fluxo Alternativo: Anfitrião rejeita hospedagem
 - a) Se o anfitrião não puder ou desejar receber o cliente, o sistema informa o fato e fornece a possibilidade de o cliente escolher outro imóvel.
 - b) Caso o cliente aceite, o caso de uso retorna para o passo 4. Caso contrário, o caso de uso é encerrado.

Fluxos de Exceção:

- I. Fluxo de Exceção: Indisponibilidade do imóvel

Se o imóvel ficar indisponível e o anfitrião não informou ao sistema antes de o cliente pesquisar as acomodações, o sistema informa o cliente que o imóvel não está disponível e cancela a solicitação, e o caso de uso retorna para o passo 2.

- II. Fluxo de Exceção: Hospede não se interessa por imóvel selecionado na busca

Se o cliente ler os detalhes do imóvel selecionado e desejar analisar as outras opções de acomodação, ele seleciona a opção voltar, o sistema retorna para a lista de imóveis com os filtros selecionados anteriormente e o caso de uso retorna para o passo 3.

- III. Fluxo de Exceção: Falta de imóveis com os filtros selecionados

Se não há oferta disponível de acomodação na região escolhida pelo hóspede, o sistema não apresentará nenhuma acomodação e o hóspede tem a opção de refinar os filtros, retornando para o passo 3, ou alterar a área de busca da acomodação, retornando para o passo 2.

Pós Condições:

- Cliente que reservou o imóvel deseja incluir mais um hóspede;
- Cliente que reservou o imóvel deseja incluir mais algum item;
- Cliente deseja saber alguma informação sobre a cidade;
- Cliente deseja saber qual o melhor trajeto para se chegar ao local;
- Anfitrião deseja saber informações sobre hóspedes.

Regras de negócio:

- RN01: O anfitrião escolhe se cobra uma porcentagem (escolhida também pelo anfitrião) do valor de entrada como garantia de reserva
- RN02: O cliente só pode se hospedar em um imóvel por vez utilizando um cadastro.

Efetuar reserva

O cliente escolhe um dos imóveis disponíveis após realizar a busca, podendo alterar ou incluir as datas de chegada e partida e a quantidade de hóspedes. Com esses dados informados, o sistema calcula o valor da hospedagem e, em alguns casos, informa a porcentagem do valor a ser paga como garantia da hospedagem. Após isso, o cliente seleciona a forma de pagamento. Caso não haja nenhum problema no pagamento e o anfitrião aceite receber o hóspede, a reserva é efetuada.

Manter Cadastro Acomodações

O usuário também pode se tornar um anfitrião, cadastrando a acomodação que deseja alugar preenchendo os campos com as informações que o sistema solicita. O anfitrião também pode alterar qualquer informação a qualquer momento, bem como retirar seu imóvel de anúncio.

Manter Serviços Adicionais

O anfitrião pode cadastrar serviços na extensão SafeService, indicando serviços locais como empregada, eletricista, encanador, empresas para aluguel de carros, mecânicos, advogados, babás, veterinários, entre outros serviços de sua confiança. A todo instante é possível alterar as informações cadastradas ou remover uma indicação.

Efetuar pagamento

O cliente efetua pagamento após efetuar a solicitação de reserva, confirmar os detalhes da hospedagem e conferir o valor a ser pago. O cliente pode abater o valor da hospedagem caso utilize um cupom de desconto. Além disso, em alguns casos o sistema informa ao cliente que ele deve pagar uma porcentagem do valor para garantir a hospedagem. Após isso, o cliente seleciona uma forma de pagamento e o sistema faz a validação dos dados no sistema de pagamento selecionado. O sistema de pagamento verifica se o cliente possui condições financeiras de fazer o pagamento e, caso seja possível, informa ao sistema Airbnb, que notifica o anfitrião que há uma reserva. Caso o anfitrião aceite receber o hóspede, ele informa ao sistema Airbnb a aprovação de reserva e o sistema Airbnb informa ao sistema de pagamento que pode ser realizado o pagamento da reserva.

Solicitar Serviços Adicionais:

Nome: Solicitar Serviços Adicionais

Identificador: CSU02

Importância:

- Risco: média
- Prioridade: alta

Sumário: Hóspede utiliza o sistema para buscar e solicitar os serviços adicionais que deseja na acomodação.

Ator Primário: Hóspede

Atores Secundários: MyUniverse, Sistema de Aluguel de Carros.

Pré-Condições: O hospede deve estar cadastrado no sistema e hospedado em alguma acomodação.

Fluxo Principal:

1. O hóspede escolhe o tipo de serviço que deseja.
2. O sistema apresenta todos os serviços que atendem ao tipo escolhido pelo hóspede.
3. Hóspede confere os serviços apresentados a partir do tipo de serviço por ele selecionado.
4. Hóspede seleciona o serviço que deseja.
5. Sistema apresenta informações detalhadas sobre o serviço selecionado pelo hóspede de acordo com a sua reserva.
6. Hóspede solicita o contrato do serviço.
7. Hóspede analisa as informações fornecidas e efetua pagamento caso tenha escolhido o cartão de crédito como a forma de pagamento.
8. Sistema confirma solicitação de serviço adicional e o caso de uso termina.

Fluxos Alternativos:

- I. Fluxo Alternativo (3): Hóspede deseja alterar o tipo de serviço que deseja.
 - a) Se o hóspede não encontrar o serviço desejado, ele tem a opção de alterar o tipo de serviço.
 - b) Caso o hóspede altere o tipo de serviço desejado, ele digita o tipo desejado e o caso de uso retorna para o passo 2.
- II. Fluxo Alternativo (5): Hóspede não se interessa por serviço selecionado
 - a) Se o hóspede ler os detalhes do serviço selecionado e desejar analisar as outras opções de serviços, ele pode selecionar a opção voltar.
 - b) Caso o hóspede selecione a opção voltar, o sistema retorna para a lista de serviços de acordo com o tipo escolhido pelo hóspede anteriormente e o caso de uso retorna para o passo 3.

- III. Fluxo Alternativo (8): Hóspede deseja efetuar pagamento em dinheiro
- a) Se o hospede desejar efetuar o pagamento em dinheiro, ele o faz diretamente com o prestador de serviço e o caso de uso retorna para o passo 9.
- IV. Fluxo Alternativo (4): Hóspede escolhe serviço indisponível.
- a) Se o hospede escolhe um serviço que já está sendo utilizado por outro usuário, o sistema mostra que está indisponível e solicita uma nova data.
- b) Caso o hóspede aceite, o mesmo informa uma nova data e o caso de uso retorna ao passo 5.

Fluxos de Exceção:

- I. Fluxo de Exceção: Falta de serviços com o tipo escolhido pelo hóspede
- Se não há nenhum serviço disponível do tipo escolhido pelo hóspede, o sistema não apresentará nenhum serviço e o hóspede tem a opção de redefinir o tipo de serviço, retornando ao passo 2.

Pós Condições:

- Hóspede contratou o serviço desejado.
- Hóspede não encontrou o serviço desejado.

Regras de Negócio:

- RN01: O hóspede pode escolher efetuar o pagamento com cartão de crédito ou com dinheiro diretamente com o prestador de serviço.
- RN02: Hóspede só pode contratar um serviço adicional se estiver hospedado em alguma acomodação.

Avaliar Acomodação

O Airbnb incentiva que os hóspedes façam comentários e avaliem os seus anfitriões e acomodações. Para isso, o sistema gera uma notificação de Comentário Pendente para a conta do usuário quando este finaliza sua estadia. Essa notificação aparecerá no menu pessoal, em Seus Comentários.

Assim que o comentário for enviado e a avaliação realizada, a notificação sumirá.

Avaliar Serviços Adicionais:

O hospede entra no aplicativo para avaliar a acomodação em que ficou e caso queira, o mesmo pode avaliar os serviços adicionais solicitados durante a sua estadia. Caso o hospede queira, poderá avaliar os serviços solicitados em outro momento.

2.2.3. Descrição dos Relacionamentos

Um relacionamento define a forma de interação entre atores, casos de uso e entre um ator e um caso de uso. Neste trabalho constatamos os seguintes relacionamentos:

- **Extend:**

Uma interação entre casos de uso que mostra uma relação de extensão entre eles. Por exemplo, a interação entre Efetuar Reserva e Buscar por Acomodação que apresenta a ideia de que o usuário pode Efetuar uma reserva após buscar por acomodação ou apenas buscar por acomodação, pois Buscar por Acomodação é um caso de uso completo.

- **Associação:**

A interação entre ator e caso de uso onde há troca de informações entre eles. Por exemplo, Usuário e Buscar Acomodação.

- **Herança:**

Uma interação entre atores que faz o mapeamento das permissões do sistema. Por exemplo, no diagrama o ator Anfitrião herda características do ator Usuário, assim um Anfitrião pode desempenhar todas as atividades que um Usuário desempenha, porém um Anfitrião tem permissões a mais que um Usuário, como: Manter Cadastro Acomodações, Manter Serviços Adicionais.

Capítulo 3 - Diagrama de Classes do Sistema

O diagrama da UML utilizado para um sistema orientado a objeto é representado pelo diagrama de classes. Neste capítulo, utilizaremos o diagrama de classes de domínio, que consiste em um conjunto de representações de classes do domínio do negócio.

A classe é subdividida em dois compartimentos, sendo o primeiro representado pelo nome da classe e o segundo pelos atributos. Cada classe pode ser relacionada umas com as outras, sendo representada por um segmento de reta que mostra o relacionamento que os objetos (objetos do mundo real) das classes têm com as outras no diagrama.

A partir da identificação das classes, dos atributos e das associações podemos construir as classes considerando as respectivas identificações e, em seguida, descrever o modelo textual desse diagrama de classes.

3.1. Diagrama de Classes de Domínio do Sistema

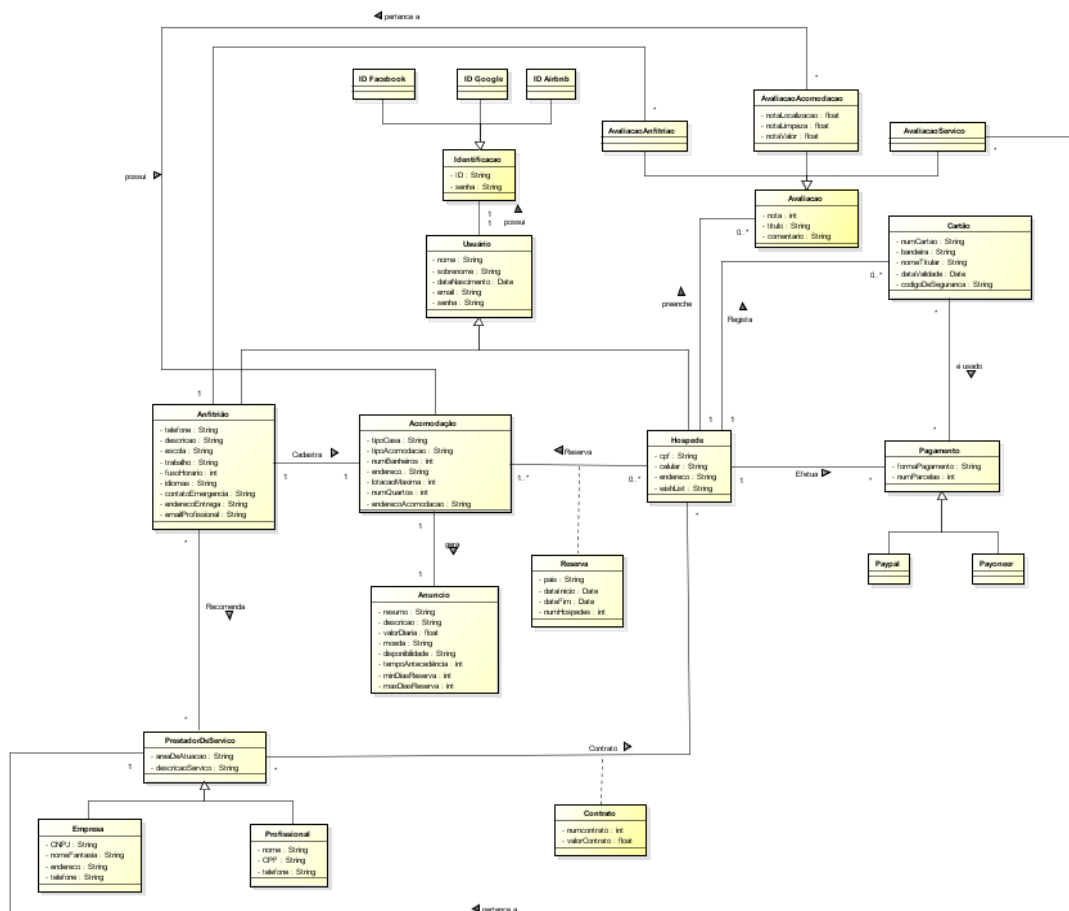


Figura 3- Diagrama de Classes do Domínio

3.2. Descrição do Diagrama de Classes do Sistema

Tendo como base a técnica de análise gramatical (Apêndice 2), que identifica possíveis classes, pegamos os casos de uso, descritos no capítulo anterior, e destacamos os nomes ou pares de nome-adjetivo. A partir disso, fizemos uma tabela contendo cada candidata a classe e as classificamos em classe ou não. Também utilizamos a técnica chamada lista de categorias (Apêndice 1), para encontrarmos outras possíveis classes que não foram identificadas pela técnica de análise gramatical. Com isso, tivemos base para construir o diagrama de classes, que será descrito nos próximos parágrafos.

A classe Usuário possui os seguintes atributos: nome, sobrenome, dataNascimento, email e senha. Estes atributos terão seus dados armazenados privadamente. Além dos atributos, possui também os seguintes relacionamentos: um usuário possui uma e somente uma identificação.

A classe Identificação possui os seguintes atributos: ID e senha.

A classe Hospede possui os seguintes atributos: cpf, celular, endereço e wishList. Estes atributos terão seus dados armazenados privadamente. Além dos atributos, possui também os seguintes relacionamentos: o Hospede herda todos os dados do Usuário, nenhum ou muitos Hospedes podem reservar uma ou muitas acomodações, um Hospede pode registrar nenhum ou muitos cartões e efetuar muitos pagamentos, e também um Hospede pode preencher nenhuma ou muitas avaliações.

A classe Reserva possui os seguintes atributos: pais, dataInicio, dataFim, numHospedes. Além disso, esta classe é uma classe associativa do relacionamento Reserva entre as classes Hospede e Acomodação.

A classe Anfitrião possui os seguintes atributos: telefone, descrição, escola, trabalho, fusoHorario, idiomas, contatoEmergencia, enderecoEntrega e emailProfissional. Além dos atributos, possui também os seguintes relacionamentos: o Anfitrião herda todos os dados do Usuário, um Anfitrião pode cadastrar nenhuma ou muitas acomodações, um ou mais Anfitriões podem recomendar também nenhum ou muitos serviços.

A classe Acomodação possui os seguintes atributos: tipoCasa, tipoAcomodacao, numBanheiros, endereço, lotacaoMAxima, numQuartos, enderecoAcomodacao. Além dos atributos, possui também os seguintes relacionamentos: uma acomodação gera um e somente um anúncio.

A classe Anuncio possui os seguintes atributos: resumo, descrição, valorDiaria, moeda, disponibilidade, tempoAntecedencia, minDiasReserva, maxDiasReserva.

A classe PrestadorDeServico possui o seguinte atributo: areaDeAtuacao. E possui o relacionamento: um ou muitos PrestadorDeServico é contratado por um ou mais Hospedes.

A classe Empresa possui os seguintes atributos: CNPJ e nomeFantasia. E a Empresa herda os dados da classe PrestadorDeServico.

A classe Profissional possui os seguintes atributos: nome e CPF. E o Profissional herda os dados da classe PrestadorDeServico.

A classe Contrato possui os seguintes atributos: numContrato e valorContrato. Esta classe é uma classe associativa do relacionamento Contrato entre as classes Hospede e PrestadorDeServico.

A classe Cartão possui os seguintes atributos: numCartao, bandeira, nomeTitular, dataValidade, codigoDeSeguranca. E possui o seguinte relacionamento: um ou muitos cartões são usados em um ou muitos pagamentos.

A classe Pagamento possui os seguintes atributos: formaPagamento e numParcelas.

A classe Avaliação possui os seguintes atributos: nota, titulo, comentário.

Capítulo 4- Diagrama de Sequência do Sistema

O diagrama de sequência do sistema é um diagrama da UML utilizado para representar as interações entre os objetos de um cenário, realizadas através de métodos ou operações. Ou seja, representa como o sistema irá trabalhar quando ele este sistema estiver em operação. Ele é construído a partir do diagrama de casos de uso, representando a sequência dos eventos desse caso de uso estendido.

Esse diagrama mostra os eventos e operações do sistema que são executados pelos atores (internos e externos) e objetos que podem realizar algum evento ou operação nesse sistema.

4.1 Diagrama de Sequência do Sistema – Fluxo Principal

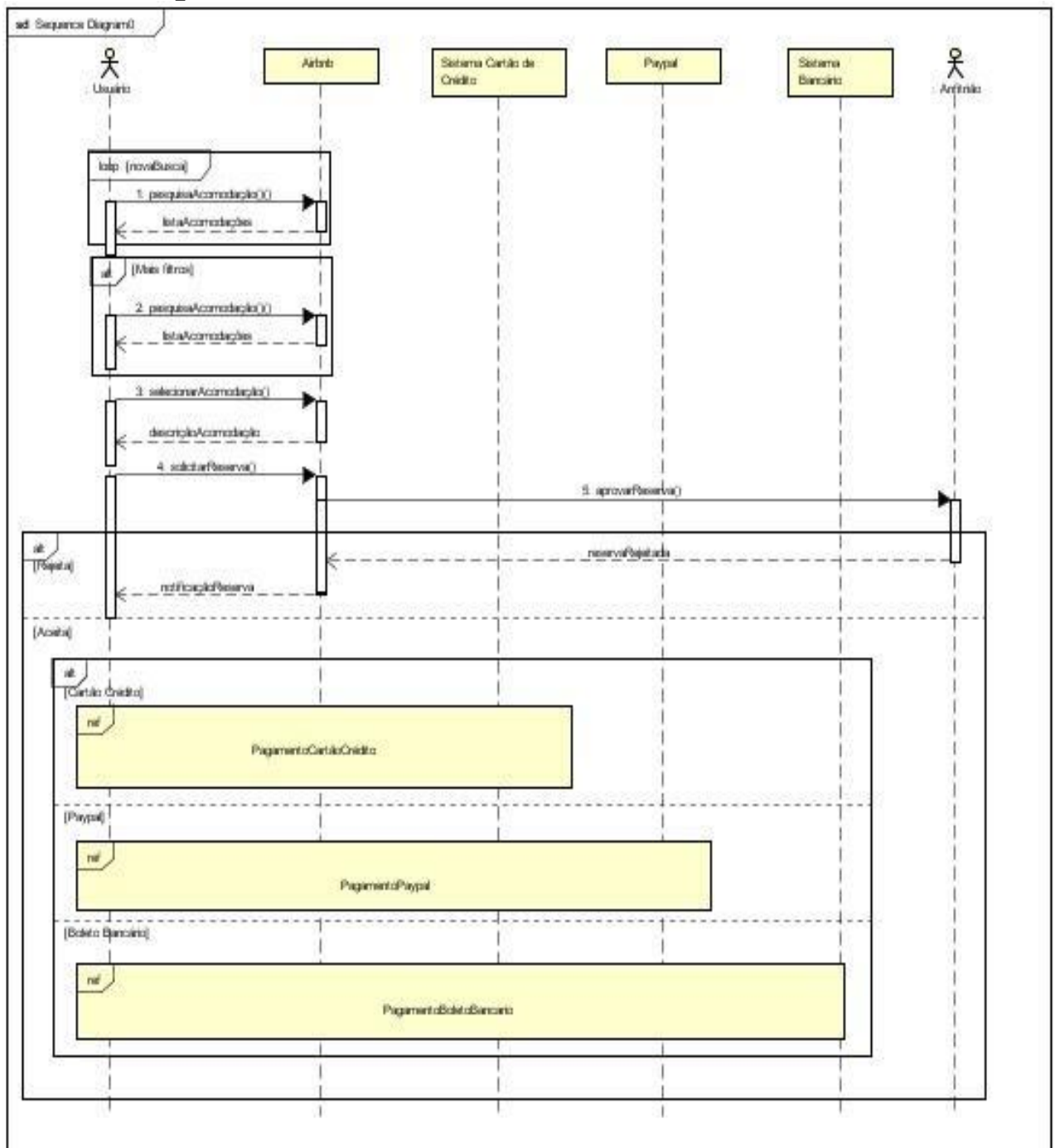


Figura 4- Diagrama de Sequencia

4.1.1. Descrição do Diagrama de Sequência do Sistema – Fluxo Principal

O diagrama de sequência ilustrado na figura 4 representa o Fluxo Principal do caso de uso estendido Buscar por Acomodação. No primeiro passo, o usuário seleciona a opção pesquisar acomodação, enviando uma mensagem do usuário ao sistema. A resposta do sistema é listar todas as acomodações que fazem parte da região escolhida pelo usuário. Além disso, o sistema apresenta ao usuário a possibilidade de adicionar filtros para refinar a pesquisa de acomodação. Após isso, o usuário marca quais filtros atendem às suas preferências e envia uma mensagem ao sistema com essas novas informações. A resposta do sistema é também de listar as acomodações, mas apenas as que atendam as preferências selecionadas pelo usuário. No passo seguinte, o usuário seleciona a acomodação que mais lhe agrada e envia outra mensagem ao sistema. Por seguinte, o usuário solicita a reserva da acomodação, enviando uma mensagem para o anfitrião, e o sistema verifica com o anfitrião se a reserva do usuário foi aprovada.

Um novo fragmento encapsulado do tipo alt é criado a fim de que decisões sejam tomadas durante o fluxo. Nele, o anfitrião recebe uma notificação sobre a reserva feita por um usuário e ele deve decidir se aceita acomodar ou não os hóspedes da reserva. Caso o anfitrião rejeite, uma mensagem do anfitrião é enviada ao sistema sobre o rejeito, e o sistema envia uma mensagem ao usuário através de uma notificação com a rejeição. Caso o anfitrião aceite, é criado outro fragmento encapsulado alt, onde o sistema apresenta as formas de pagamento disponíveis para realizar a reserva.

4.2. Diagrama de Sequência do Sistema – Pagamento Cartão de Crédito

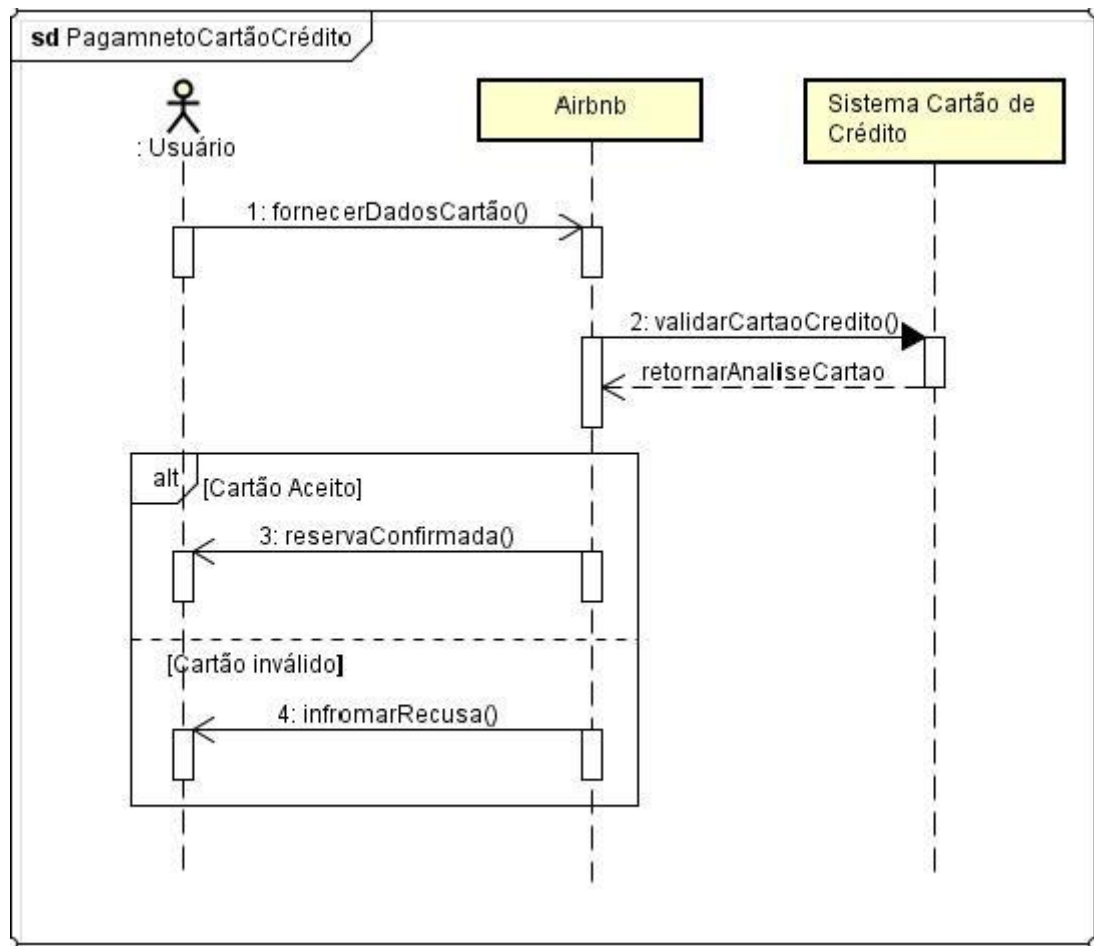


Figura 5- Diagrama de Sequencia Pagamento Cartao de Crédito

4.2.1. Descrição do Diagrama de Sequência do Sistema – Pagamento Cartão de Crédito

O diagrama de sequência ilustrado na figura 5 representa o Fluxo de Pagamento via cartão de crédito. No primeiro passo, o usuário fornece os dados do cartão de crédito, enviando uma mensagem ao sistema. O sistema envia uma mensagem ao Sistema de Cartão de Crédito para validar os dados fornecidos pelo usuário, e o Sistema de Cartão de Crédito responde ao sistema com o retorno da análise do cartão.

Um novo fragmento encapsulado do tipo alt é criado a fim de que decisões sejam tomadas durante o fluxo. Nele, o sistema de cartão de crédito analisa se os dados fornecidos pelo usuário são os mesmos que estão registrados em nome de tal usuário. Em caso positivo, o cartão é

aceito e o sistema envia uma mensagem para o usuário confirmando a reserva da acomodação. Em caso negativo, o sistema envia uma mensagem ao usuário informando que os dados do cartão informados são inválidos e que a reserva da acomodação foi recusada.

4.4. Diagrama de Sequência do Sistema – Pagamento PayPal

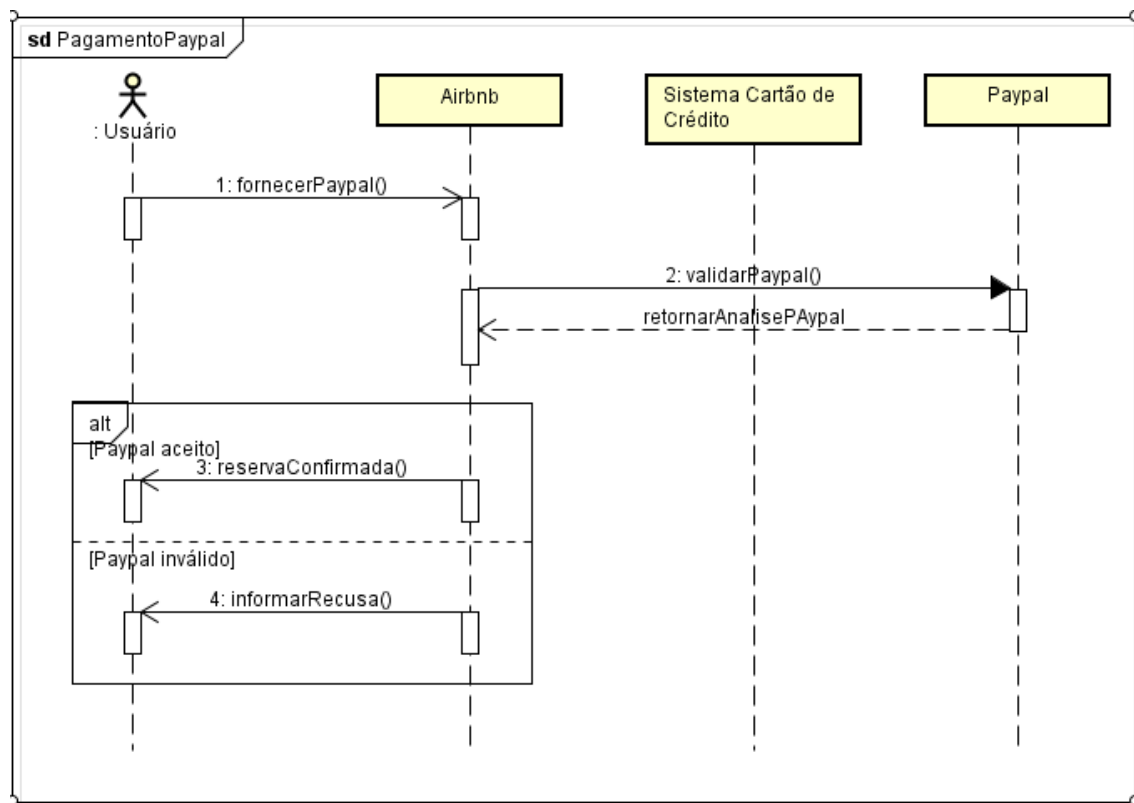


Figura 6- Diagrama de Sequencia Pagamento PayPal

4.3.1. Descrição do Diagrama de Sequência do Sistema – Pagamento Paypal

O diagrama de sequência ilustrado na figura 6 representa o Fluxo de Pagamento via cartão Paypal. No primeiro passo, o usuário fornece os dados do seu Paypal, enviando uma mensagem ao sistema. O sistema envia uma mensagem ao Paypal para validar os dados fornecidos pelo usuário, e o Paypal responde ao sistema com o retorno da análise do cartão.

Um novo fragmento encapsulado do tipo alt é criado a fim de que decisões sejam tomadas durante o fluxo. Nele, o Paypal analisa se os dados fornecidos pelo usuário são os mesmos que estão registrados em nome de tal usuário. Em caso positivo, o Paypal é aceito e o sistema envia uma

mensagem para o usuário confirmando a reserva da acomodação. Em caso negativo, o sistema envia uma mensagem ao usuário informando que os dados Paypal informados são inválidos e que a reserva da acomodação foi recusada.

4.4. Diagrama de Sequência do Sistema – Pagamento Boleto Bancário

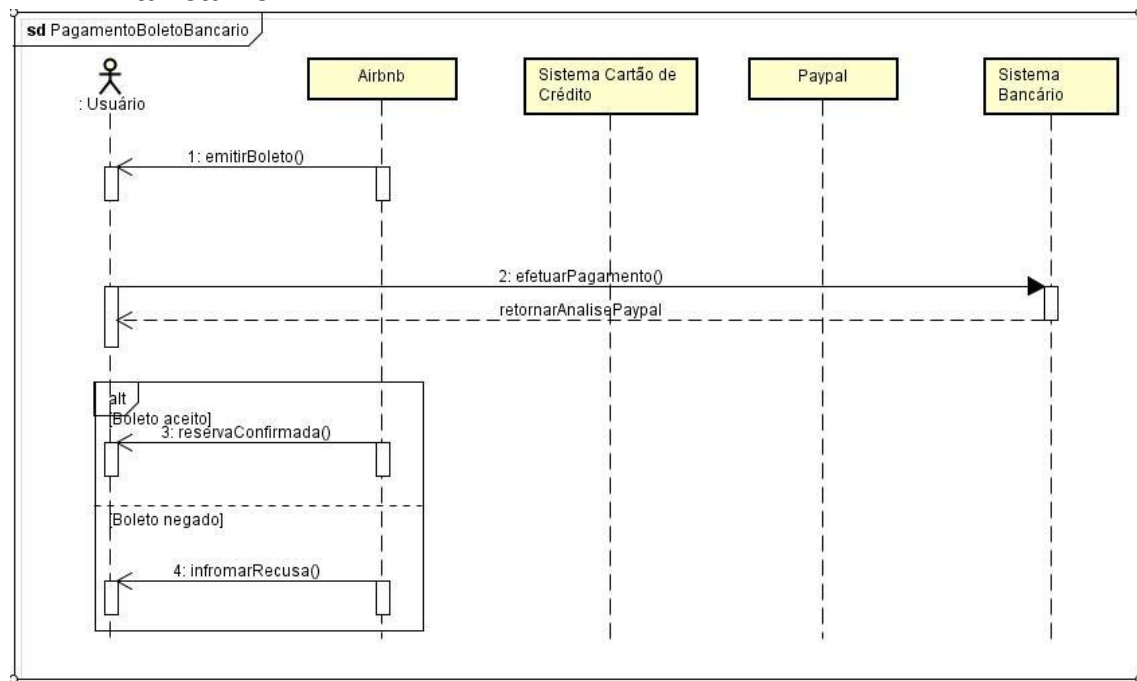


Figura 7- Diagrama de Sequencia Pagamento Boleto Bancário

4.4.1. Descrição do Diagrama de Sequência do Sistema – Pagamento Boleto Bancário

O diagrama de sequência ilustrado na figura 7 representa o Fluxo de Pagamento via boleto bancário. No primeiro passo, o sistema emite o boleto ao usuário, enviando uma mensagem ao usuário. Por seguinte, o usuário efetua o pagamento do boleto bancário, enviando uma mensagem do usuário ao sistema bancário. A resposta do sistema bancário é a de retornar a análise de crédito do usuário.

Um novo fragmento encapsulado do tipo alt é criado a fim de que decisões sejam tomadas durante o fluxo. Nele, o sistema bancário analisa se o usuário que efetuou o pagamento desse boleto possui condições financeiras de realizar tal pagamento. Em caso positivo, o sistema informa

ao usuário que o boleto foi aceito e confirma a reserva do usuário, enviando uma mensagem ao usuário. Em caso negativo, o sistema informa ao usuário que o boleto foi rejeitado e recusa a reserva da acomodação, enviando uma mensagem ao usuário

Capítulo 5- Diagrama de Pacotes

O Diagrama de Pacotes é um diagrama da UML que apresenta os pacotes, agrupados de acordo com a sua especificidade e suas dependências. Este diagrama é utilizado para ilustrar a arquitetura lógica de um sistema, mostrando o agrupamento de suas classes, e pode ser utilizado em qualquer fase do processo de modelagem. Um pacote é um mecanismo para organizar elementos de um modelo(classes, diagramas ou até mesmo outros pacotes) em grupos. Os pacotes possuem uma relação de dependência com outros pacotes. Esse diagrama visa organizar os modelos e é representado por uma pasta com o nome no interior ou na pega.

O diagrama é estruturado por camadas, onde cada camada é composta por um agrupamento de classes ou pacotes, que são representados por uma divisão horizontal, conhecidas como partições horizontais. A divisão é feita por camadas que se organizam em Interface com Usuário (IU), Aplicação, Domínio, Infraestrutura do Negócio e Serviços Técnicos.

5.1. Diagrama de Pacotes

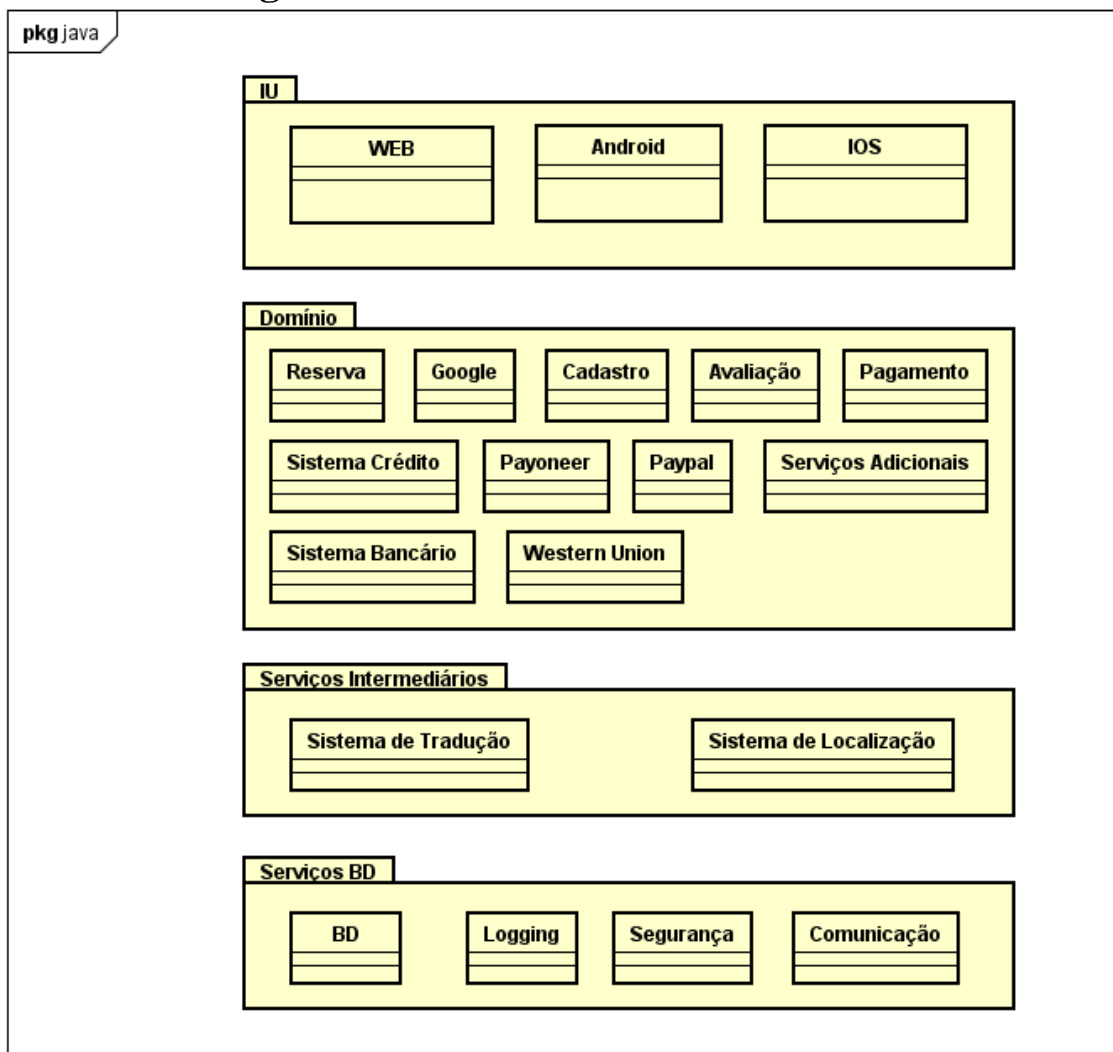


Figura 8- Diagrama de Pacotes

5.1.1. Descrição do Diagrama de Pacotes

O diagrama de pacotes divide está dividido de acordo com as suas interfaces, cuja cada interface possui as suas classes relacionadas.

Na IU (Interface do Usuário), temos as classes onde o usuário pode acessar o sistema, manipulando de acordo com as permissões concedidas ao usuário. Já no domínio, estão as classes do sistema que possuem são mais necessidade de reuso e de maior impacto de mudanças. Já nos Serviços Intermediários estão as classes que possuem uma certa importância, mas que não são utilizadas tanto quanto as classes do domínio. E no pacote Serviços BD são as classes que precisam de um banco de dados para armazenar os dados das outras classes dos outros pacotes.

Capítulo 6

Erro! Fonte de referência não encontrada.

Este capítulo apresenta a transformação do diagrama de classes do sistema em um diagrama de classes do projeto. Neste caso, o sistema não é apresentado como uma caixa preta, mas sim seu funcionamento interno. Neste diagrama, são criadas todas as classes que participam do software, através da análise do diagrama de sequência do projeto que serão adicionadas ao diagrama de classes de análise feito anteriormente, além da inclusão dos atributos, métodos, associações e setas de associação e relações de todas as classes deste diagrama.

A representação do diagrama de classes do projeto se assemelha com o diagrama de classes de análise, porém difere pois as classes da fase de projeto são classes de software, além de uma entidade representar um software, anteriormente representada como um conceito do domínio do problema.

6.1 Diagrama de Classes de Projeto

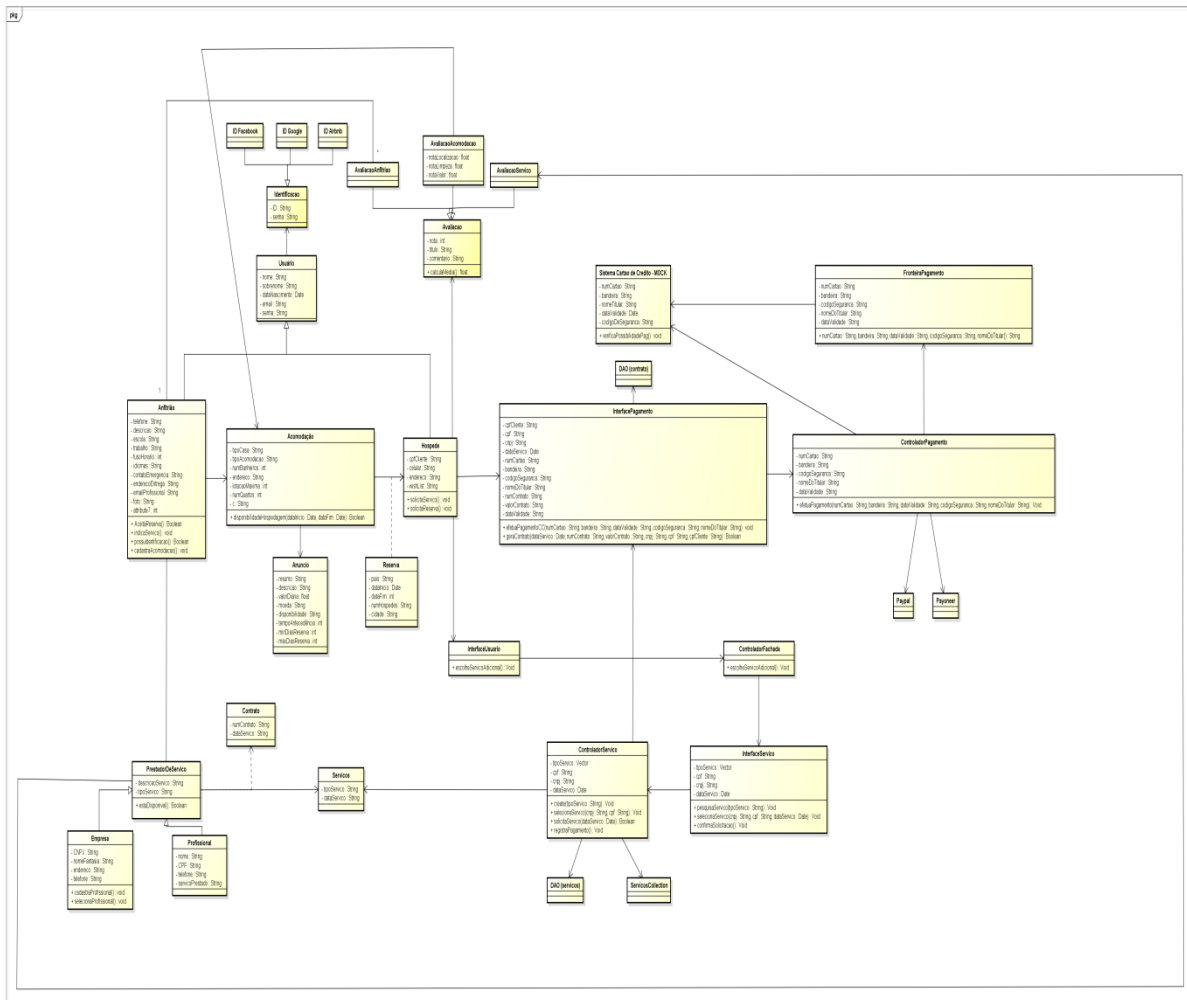


Figura 9- Diagrama de Classes do Projeto

6.1.1 Descrição do Diagrama de Classes de Projeto

O diagrama de classes do projeto é utilizado para ilustrar as principais classes do sistema a ser desenvolvido, demonstrando os métodos, ajudando a definir o nicho das classes e, por sua vez, contribuindo para uma maior organização do projeto no quesito coesão.

Ainda, através desse diagrama, são apresentados os relacionamentos entre as classes, que podem ser de herança ou outra forma de dependência. Nesse caso, é possível de forma geral, entender como o código irá ser desenvolvido e quais será a estrutura principal do mesmo.

Dentre os relacionamentos existentes neste diagrama de classes, são apresentados o de herança, que influencia diretamente no conteúdo que as classes filhos irão possuir. Nessa relação, existe necessariamente uma classe “Pai” e indefinidas classes “Filho”.

Há ainda a relação de interface, onde uma classe deve conter classes abstratas e seus herdeiros deverão obrigatoriamente implementar todas as funções contidas na classe interface. Ela se diferencia do relacionamento anterior, pois o que será “herdado” são as classes a serem implementadas e não os métodos já definidos.

Capítulo 7– Diagrama de Sequência do Projeto

Este capítulo apresenta a transformação do diagrama de sequência do sistema em um diagrama de sequência do projeto. Nesse caso, o sistema não é representado como uma caixa preta, mas sim seu funcionamento interno.

Neste diagrama, são criados diferentes objetos que podem ser nomeados ou uma coleção. Objeto nomeado é aquele ao qual foi dado um nome. Normalmente um objeto nomeado é utilizado quando precisa ser referenciado em mais de um lugar. E objeto em uma coleção é aquele ao qual foi dado um nome com elemento seletor. Nesse caso, o nome do objeto é separado do nome de sua classe por um sinal de dois pontos.

Alguns tipos de mensagens representados neste diagrama são semelhantes as que foram apresentadas no diagrama de sequência do sistema, sendo eles: síncrona, assíncrona, de retorno e reflexiva. Além desses tipos, utilizamos uma mensagem do tipo criação que é representada por setas com linhas tracejadas, cuja linha de vida inicia na posição em que é criado, além de apresentar o estereótipo <<create>> na seta.

7.1. Diagrama de Sequência Fluxo Principal e do Fluxo Alternativo 4

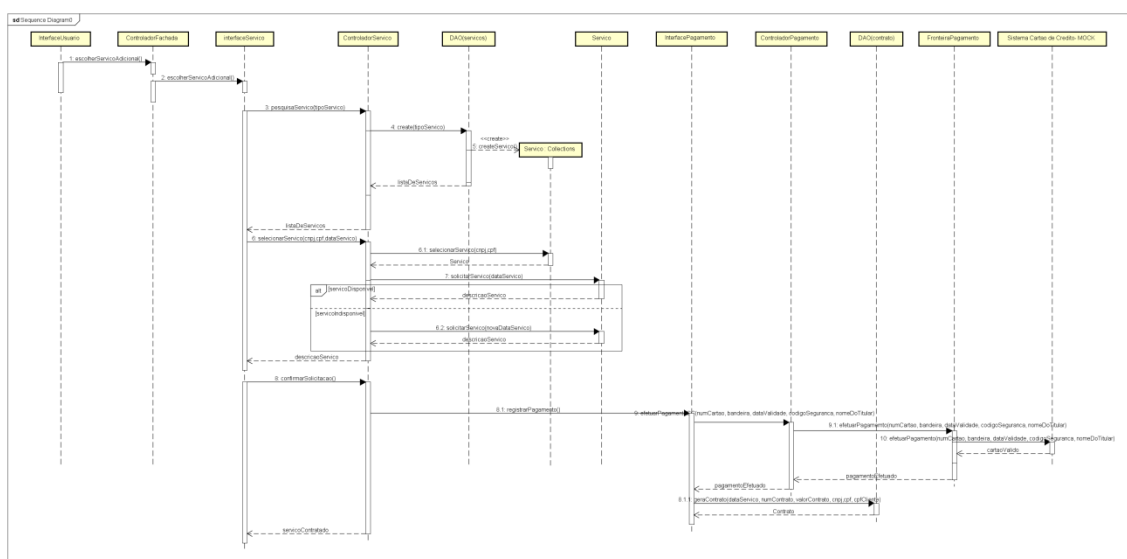


Figura 10- Diagrama Sequência de Projeto

7.1.1. Descrição do Diagrama de Sequência de Projeto

Este diagrama representa o fluxo principal do caso de uso Solicitar Serviços Adicionais e o fluxo alternativo: Hospede escolhe serviço já contratado. Para começar, a InterfaceUsuário disponibiliza ao usuário a opção de escolher um serviço adicional. Quando o usuário selecionar essa opção, uma mensagem é enviada para o ControladorFachada e em seguida para a interfaceServico, assim o usuário é redirecionado para a interface de pesquisa de serviço adicional. Nesta interface, o usuário escolhe o tipo de serviço que deseja, com isso é enviada uma mensagem para o ControladorServico e depois o mesmo envia uma mensagem para o DAO(Data Access Object), ferramenta que faz a ligação entre o controlador e a base de dados, solicitando a lista de serviços do tipo desejado. O DAO cria uma coleção de serviços na base de dados e retorna a lista de serviços desejada para o ControladorServico e o mesmo mostra esta lista para o usuário na interfaceServico. Em seguida, a interfaceServico envia uma mensagem para o ControladorServico passando como parâmetro a identificação do serviço desejado(cnpj ou cpf) e a data desejada. O controladorServico envia uma mensagem para a coleção de serviços que foi criada na base de dados solicitando um objeto serviço de acordo com os parâmetros passados. Após isso, o ControladorServico irá se comunicar diretamente com este objeto serviço, enviando uma mensagem de solicitação do mesmo passando como parâmetro a data desejada. Nesta parte, temos duas opções: primeiro se o serviço estiver disponível é retornado para o ControladorServico a descrição detalhada do serviço. Caso o serviço esteja ocupado (Fluxo Alternativo) o usuário tem a opção de escolher uma nova data para o serviço, escolhendo esta opção é enviada uma mensagem como uma nova data para o objeto serviço, e uma mensagem de retorno é enviada para o controlador com a descrição detalhada do serviço. O controlador envia a descrição detalhada para a interfaceServico. Assim, o usuário lê a descrição e envia uma mensagem para o ControladorServico confirmando a solicitação do serviço. O ControladorServico envia uma mensagem para interfacePagamento e o usuário é redirecionado para a interface onde efetuará o pagamento. Nesta interface, o usuário envia uma mensagem com os seguintes parâmetros:

numero do cartão, bandeira, data de validade, código de segurança, nome do titular para o controlador de pagamento. O ControladorPagamento envia uma mensagem para a fronteiraPagamento com os mesmos parâmetros listados anteriormente. E esta fronteira fica responsável por se comunicar com o sistema de cartão de crédito para validar o pagamento. O sistema de cartão de crédito envia uma mensagem de retorno para a fronteiraPagamento validando o cartão e a mesma envia uma mensagem de retorno para o controladorPagamento confirmando o pagamento e este repassa essa mensagem para a interfacePagamento. O usuário finaliza seu pagamento e envia uma mensagem para gerar um contrato para o DAO de contrato com os parâmetros: dataServico, numContrato, valorContrato, cnpj,cpf, cpfCliente. E este envia uma mensagem de retorno com o contrato. E por fim, o controladorServico envia uma mensagem para a interfaceServico informando que o serviço foi contratado.

Capítulo 8- Diagrama de Atividade

O diagrama de atividade é um diagrama da UML representa os fluxos conduzidos por processos, que tem como objetivo mostrar o fluxo de controle atividade para outra dentro de um processo. Este diagrama é essencialmente um gráfico de fluxo que ilustra onde serão empregadas as atividades para fazer a modelagem de aspectos dinâmicos do sistema e mostra como uma atividade depende da outra dentro do processo. Este diagrama é composto por:

- Atividades: fluxo de trabalho que deve ser realizado;
- Sub-atividade: execução de uma sequência não atômica de atividades;
- Transição: fluxos entre atividades, que mostram a dependência entre essas atividades;
- Ação: etapa na atividade na qual um agente executa uma tarefa;
- Decisão: mostra as diferentes transições caso uma condição seja atendida. Tem uma entrada e de duas ou mais saídas. Um token de entrada surge apenas em uma das saídas;
- Raia: regiões associadas a um objeto do modelo que diferencia as unidades organizacionais;
- Bifurcação (Fork): divide um único fluxo em fluxos simultâneos executados ao mesmo tempo;
- Sincronização (Join): Concatenação dos fluxos simultâneos vindos do Fork;
- Objetivo: objetivo da atividade;
- Enviar sinal de ação: Ação que envia uma mensagem ou um sinal para outra atividade, a um thread na mesma atividade ou a um meio externo;
- Recepção do sinal: Recepção do envio;
- Região: Agrupamento de uma ou mais atividades;
- Exceção: Atividades que ocorrem em decorrência de uma exceção.

Este diagrama representa o processo de Busca Por Serviços Adicionais. No início desse processo, o Hóspede realiza a atividade de escolher o tipo de serviço desejado. Essa atividade realizada no mínimo uma vez, até que a condição do Hóspede seja atendida. Enquanto não for atendida, a resposta no Fork de abertura será retornar à atividade anterior, mas se for atendida, a resposta é uma nova transição de atividade que sejam apresentados os serviços que atendam ao tipo selecionado pelo Hóspede anteriormente. Em seguida, ocorre outra transição para a atividade do Hóspede de selecionar, dentre todos os serviços apresentados, o que atenda às suas condições. Esta atividade está na área de interrupção, ou seja, o processo pode ser encerrado enquanto alguma das atividades que estão na área de interrupção for interrompida. Após isso, ocorre outra transição para a atividade do SafeService de apresentar informações detalhadas sobre o serviço selecionado. Essa atividade está fora da área de interrupção. Depois, ocorre outra transição para inserção da data desejada. Atividade essa realizada pelo Hóspede e que está na área de interrupção. Essa

atividade é levada por outra transição a outro Fork, cuja condição é a data selecionada pelo Hóspede estar disponível. Se a condição não for atendida, ocorre uma transição de retorno para a atividade de inserir a data desejada, mas se a condição da data for atendida ocorre uma transição para a próxima atividade, que é a da confirmação da solicitação do contrato de serviço. Essa atividade está na área de interrupção. A próxima transição leva a outro Fork de abertura, onde será decidida a forma de pagamento. Se o pagamento for realizado em dinheiro, ocorre uma transição para um Fork de encerramento, onde o pagamento em dinheiro é realizado por quem já havia decidido por essa opção e por quem havia decidido o pagamento por cartão de crédito e teve o pagamento aprovado pelo SafeService e está fora da área de interrupção. Esse Fork leva a atividade de confirmar a solicitação do serviço e leva a uma transição que finalizará o processo. Essa atividade é realizada pelo SafeService. Caso o Fork de abertura seja decidido pelo uso do cartão de créditos, ocorre uma transição para a atividade do Hóspede de informar os serviços bancários. Essa atividade está dentro da área de interrupção. Após isso, ocorre uma transição para a validação dos dados fornecidos pelo Hóspede, atividade essa realizada pelo sistema de cartão de créditos. Essa atividade leva a uma transição para um Fork de abertura, cuja condição é a de o cartão será aprovado pelo SafeService. Caso a condição seja atendida, ocorre uma atividade para o Fork de encerramento de aprovação do pagamento, que foi descrito anteriormente, mas caso a condição não seja atendida, ocorre uma transição para a atividade de informar recusa, que leva ao encerramento do processo.

Capítulo 9- Diagrama de Estados

O diagrama de transição de estados é um diagrama da UML que representa o comportamento do estado de um objeto no decorrer da execução de processos de um sistema. Esse estado depende da atividade na qual ele está processando. Com isso, o objeto varia o seu comportamento durante as transições, responsáveis pelas mudanças de estado. Este diagrama é composto por:

- Estado: condição ou situação na vida de um objeto, satisfazendo algumas condições e executando algumas atividades;
- Transição: avanço de um estado do objeto para outro. São representados por caminhos no diagrama;
- Condição: causa necessária para que haja uma transição de estado. Decorre da ocorrência de um evento associado a ocorrência de uma transição;
- Evento: nome de um sinal, mensagem ou notificação recebida por um objeto que torna a transição habilitada;
- Estado inicial: estado por onde se inicia a leitura do diagrama;
- Estado final: estado que representa o fim de uma máquina;
- Barra de sincronização: separa uma transição em várias transições executadas ao mesmo tempo;
- Estado composto: estado constituído de um conjunto de sub-estados;
- Sincronização: permite que o clock de dois ou mais processos paralelos estejam sincronizados em um determinado momento do processo;
- Ação: atividade do sistema que efetua a transição de estado.

9.1 Diagrama de estados

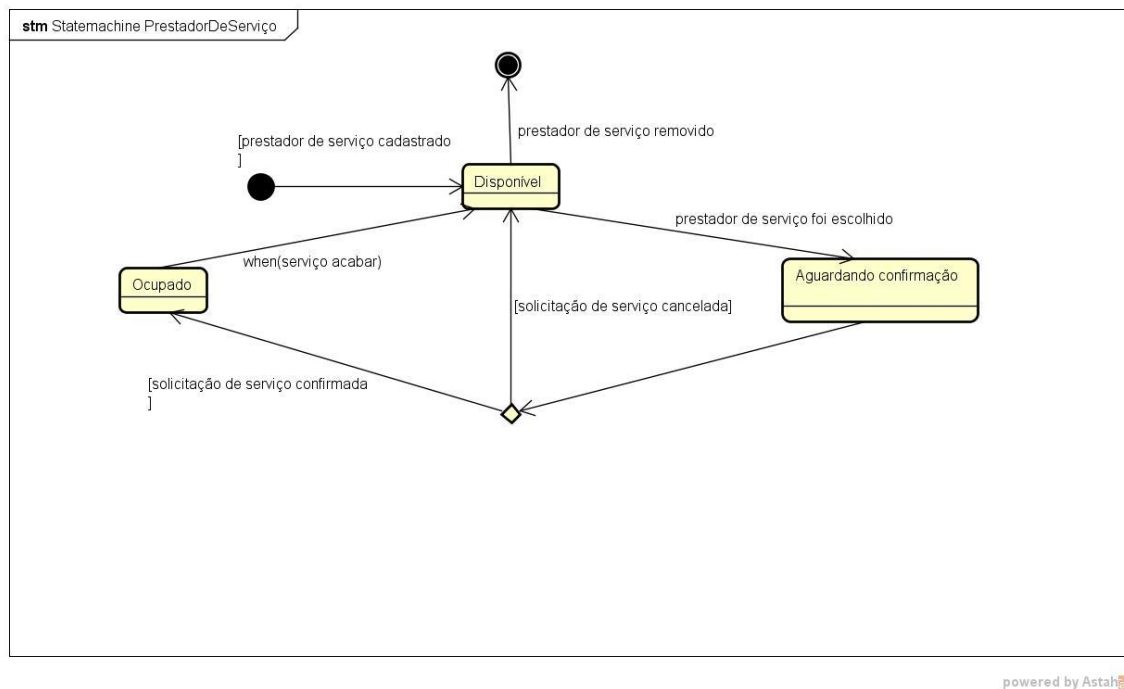


Figura 11- Diagrama de Estados

9.1.1 Descrição do Diagrama de Estados

Este diagrama representa o objeto da classe PrestadorDeServiço. No início, o prestador de serviço cadastrado fica no estado Disponível. Após isso, ocorre uma transição de estado onde o prestador de serviço é escolhido por um objeto de Hóspede e altera o seu estado para Aguardando confirmação. Após isso, uma nova transição é apontada a um nó de decisão. Caso a condição estabelecida de confirmação for atendida, o prestador de serviço altera seu estado para Ocupado após a transição de solicitação de serviço confirmada. Caso a condição não seja atendida, o prestador de serviço volta a ficar no estado Disponível. No estado Ocupado, o prestador de serviço realiza o seu trabalho e, só quando o serviço é terminado, uma transição de estado retorna o estado desse prestador para Disponível.

Caso o prestador de serviço esteja no estado Disponível e seja removido, o objeto PrestadorDeServiço encerra a sua vida.

Apêndice 1 – Lista de Categorias

Categoria ou classe conceitual	Exemplo
Transação de negócio (envolvem dinheiro)	Pagamento, Reserva, Aluguel, Contrato
Transação de item	AluguelAcomodação, Serviço, Profissional
Produtos ou serviços relacionados a transação ou item	Lugar, ListaTelefônica, GuiaViagem, Mapa
Onde a transação é registrada	No próprio sistema
Papeis de pessoas ou organizações	Hóspede, Banco, Anfitrião, AdministradoraCartãoCrédito, Profissional
Local de transação ou serviço	Aplicativo, Site, Acomodação
Eventos notáveis	Pagamento, Reserva, Contrato
Descrição de coisas	DescriçãoAcomodação
Catálogos	CatálogoAcomodação
Contêiner de coisas (físico ou informação)	InformaçõesAcomodação
Outros sistemas	SistemaDeAutorizaçãoDeCrédito, SistemadeLocalizaçãoGPS
Registros de finanças, trabalho, contrato, assuntos legais	Comprovante, NotaFiscal
Instrumentos financeiros	Western Union, PayPal, Cartão de Crédito
Horários, manuais, documentos, consultados para realizar trabalho	ListaDeAlteraçãoDePreço

Apêndice 2 – Análise Gramatical

Nome	Significado	É uma classe	Justificativa
Aplicativo	Aplicativo Airbnb, onde o hóspede navega no sistema Airbnb.	Não	Consiste no próprio sistema.
Comunicação	Modo onde o hóspede se comunica com o prestador de serviços.	Não	Relacionamento Hóspede/Anfitrião ou Hóspede/Serviço
Mensagem	Mensagem de texto que o hóspede envia para o prestador de serviços.	Não	Comunicação Usuário/Prestador de Serviços
Acomodação	Local onde o hóspede está hospedado.	Sim	O principal conjunto de dados gerenciado pelo sistema.
Anfitrião	Pessoa que recebe os hóspedes durante a estadia, além de indicar alguns serviços.	Sim	Gerencia todas as informações de cada anfitrião.
Cadastro	Conjunto de informações pessoais do usuário.	Não	Faz parte do escopo do sistema.
Hóspede	Pessoa que faz reserva de uma acomodação.	Sim	Gerencia as informações de cada hóspede.
Reserva	Confirmação do anfitrião para o hóspede sobre sua solicitação de hospedagem.	Sim	Gerencia as informações de cada hospedagem.
Contato	Meio utilizado para comunicação entre o hóspede e o anfitrião.	Não	É um tipo de comunicação do sistema
Comentários	Mensagem de texto que os hóspedes anteriores utilizam para escrever sobre sua experiência na acomodação e da área onde ela	Não	Um tipo de comunicação do sistema

	está localizada.		
Serviço	Refere-se a necessidade do hóspede para atender ao problema ocorrido.	Sim	Gerencia todas as informações de cada um dos serviços.
Estadia	Refere-se ao tempo em que o hóspede ficou na acomodação.	Não	Atributo de Reserva ou Contrato
Contrato	Acordo a ser entre cumprido diante de determinadas condições.	Sim	Gerencia todas as informações de cada um dos contratos.
Região	Local onde se localiza a acomodação.	Não	Não está no limite do sistema
Pagamento	Meio de remuneração sobre a reserva da acomodação ou prestação de serviço.	Sim	Gerencia as informações de cada pagamento
Modo de Pagamento	Apresenta uma gama de opções de pagamento, cada uma com seu sistema independente.	Não	Está contido na classe Pagamento.
Profissional	Pessoa que realiza um serviço.	Sim	Gerencia todas as informações de cada profissional.
Prestador de Serviços	Modo de realização de um serviço.	Sim	Gerencia todas as informações de cada prestador de serviços
Localização	Região onde o hóspede deseja se acomodar.	Não	Não está no limite do sistema.
Carros para aluguel	Lista de carros que o hóspede pode alugar durante sua estadia na acomodação.	Não	Não está no limite do sistema.
MyUniverse	Serviço de tradução utilizada pelo hóspede durante sua estadia.	Não	Não está no limite do sistema.
Conta	Informações preenchidas pelo usuário.	Não	Está contida na classe Cadastro.
Validação	Confirmação dos requisitos funcionais do sistema.	Não	Está contida na classe Cadastro.

Levantamento	Recolhimento dos requisitos funcionais do sistema.	Não	Está contida na classe Cadastro.
Campos	Áreas de seleção de filtros.	Não	Simplesmente um atributo de Acomodação
Filtro	Conjunto de preferências do hóspede sobre uma acomodação.	Não	Simplesmente um atributo de Acomodação
Prazo de não pagamento de multa	Tempo em que o hóspede deve pagar para não ter um acréscimo de uma taxa.	Não	Simplesmente um atributo de Pagamento.
Multa	Valor a ser pago caso alguma infração seja cometida.	Não	Simplesmente um atributo de Pagamento.
Todo instante	Período em que o hóspede pode indicar novos serviços.	Não	Somente uma variação de tempo.
Cupom de desconto	Vale desconto para abater o valor a ser pago pela estadia.	Não	Simplesmente um atributo de Pagamento.
Condições financeiras	Conjunto de regras estabelecidas para a realização do pagamento.	Não	Atributo de Pagamento
Palavra-passe	Combinação de caracteres para o acesso de usuário à conta no Airbnb.	Não	Atributo de Cadastro
Avaliação	Combinação de comentário, nota e título que os hóspedes antigos dão à uma acomodação.	Sim	Gerencia todas as informações de cada avaliação.
Paypal	Sistema que o usuário realiza o pagamento no Airbnb.	Não	Não está no limite do sistema
Western Union	Sistema que o usuário realiza o pagamento no Airbnb.	Não	Não está no limite do sistema
Google Wallet	Sistema que o usuário realiza o pagamento no Airbnb.	Não	Não está no limite do sistema
Facebook	Sistema com o qual o usuário vincula ao Airbnb para se cadastrar.	Não	Não está no limite do sistema
Google	Sistema com o qual o usuário vincula ao Airbnb para se cadastrar.	Não	Não está no limite do sistema.

Permissões	O que um usuário tem acesso de acordo com a sua função no sistema.	Não	Não está no limite do sistema.
------------	--	-----	--------------------------------

Apendice 3 – Classes Java

Classe IdGoogle:

```
public class IdGoogle extends Identificacao{  
}
```

Classe IdAirbnb:

```
public class IdAirbnb extends Identificacao{  
}
```

Classe IdFacebook:

```
public class IdFacebook extends Identificacao{  
}
```

Classe Identificação:

```
public class Identificacao{  
    protected String id, senha;  
}
```

Classe Usuário:

```
public class Usuario{  
    protected String nome, sobrenome, email, senha;  
    protected Date dataNascimento;  
}
```

Classe Anfitrião:

```
public class Anfitriao extends Usuario{  
    private String telefone, descricaoPessoal, localFormacaoEstudantil, empresa, profissao;  
    private String idiomas, contatoEmergencia, enderecoEntrega, emailProfissional ;  
    private int fusoHorario;  
  
    public void Anfitriao(String telefone, String descricaoPessoal, String localFormacaoEstudantil, String empresa, String profissao,  
        String idiomas, String contatoEmergencia, String enderecoEntrega, String emailProfissional,int fusoHorario){  
  
        super();  
        this.telefone = telefone;  
        this.descricaoPessoal = descricaoPessoal;  
        this.localFormacaoEstudantil = localFormacaoEstudantil;  
        this.empresa = empresa;  
        this.profissao = profissao;  
        this.idiomas = idiomas;  
        this.contatoEmergencia = contatoEmergencia;  
        this.enderecoEntrega = enderecoEntrega;  
        this.emailProfissional = emailProfissional;  
        this.fusoHorario = fusoHorario;  
    }  
}
```

Classe Hospede:

```
public class Hospede extends Usuario{  
    private String cpf, celular, endereco;  
    private ArrayList<String> wishlist;  
}
```

Classe Prestador de Servico:

```
public class PrestadorDeServico{  
    protected String areaDeAtuacao, descricaoServico;  
}
```

Classe Empresa:

```
public class Empresa extends PrestadorDeServico{  
    private String CNPJ, nomeFantasia, endereco, telefone;  
}
```

Classe Profissional:

```
public class Profissional extends PrestadorDeServico{  
    private nome, CPF, telefone;  
}
```

Classe Pagamento:

```
public class Pagamento{  
    protected String formaPagamento;  
    protected int numParcelas;  
}
```

Classe Paypal:

```
public class PayPal extends Pagamento{  
}
```

Classe Paioneer:

```
public class Paioneer extends Pagamento{  
}
```


Classe Avaliação:

```
public class Avaliacao{  
    protected String nota, titulo;  
    protected int nota;  
}
```

Classe Avaliação Acomodação:

```
public class AvaliacaoAcomodacao extends Avaliacao{  
    protected float notaLocalizacao, notaLimpeza, notaValor;  
},
```

Classe Avaliação Serviço:

```
public class AvaliacaoServico extends Avaliacao{  
}
```

Classe Avaliação Anfitrião:

```
public class AvaliacaoAnfitriao extends Avaliacao{  
}
```

Classe Contrato:

```
public class Contrato{  
    protected int numcontrato;  
    protected float valorContrato;  
}
```

Classe Reserva:

```
public class Reserva{  
    protected int numHospedes;  
    protected String pais;  
    protected Date dataInicio, dataFim;  
}
```

Classe Cartão:

```
public class Cartao{  
    protected String numCartao, bandeira, nomeTitular, codigoDeSeguranca;  
    protected Date dataValidade;  
}
```

Classe Acomodação:

```
public class Acomodacao{  
    protected String tipoCasa, tipoAcomodacao, endereco, enderecoAcomodacao;  
    protected int numBanheiros, lotacaoMaxima, numQuartos;  
}
```

Classe Anuncio:

```
public class Anuncio{  
    protected String resumo, descricao, moeda, disponibilidade;  
    protected int tempoAntecedencia, minDiasReserva, maxDiasReserva;  
    protected float valorDiaria;  
}
```