

## Exemplo de trabalho de 2015.1

Este exemplo é para dar uma ideia do que é esperado em termos de estrutura e apresentação.

Observe que o trabalho está bom, mas não está perfeito e contém erros. Um erro muito importante (do "includes" no diagrama de casos de uso) foi sinalizado. Os outros erros não foram apontados. Portanto, cuidado! O seu objetivo é entregar um trabalho melhor ainda!

Importante: As instruções do trabalho mudaram! Baseiem-se nas instruções e nas aulas desse semestre! Não tentem fazer um paralelo entre os diagramas de vocês e os diagramas do trabalho de exemplo.

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

## Extensão do HelloUniverse: *MyUniverse*

Beatriz Carvalho  
Rafaela Brum  
Suelen Regina  
Thainá Figueiredo

## Sumário

Resumo	1
Capítulo 1 - Introdução	2
1.1 - HelloUniverse	2
1.2 - MyUniverse	2
1.3 - Diagrama de Contexto	3
Capítulo 2 - Diagrama de Casos de Uso	5
2.1 - Diagrama de Casos de Uso do Sistema	5
2.2 - Descrição do Diagrama de Casos de Uso	5
2.3 - Descrição dos Atores e Casos de Uso	6
2.3.1 - Descrição dos atores	6
2.3.2 - Descrição dos casos de uso	7
Capítulo 3 - Diagrama de Classes de Domínio	12
3.1 - Diagrama de Classes de Domínio do Sistema	12
3.1.1 - Descrição do Diagrama de Classes de Domínio	12
Capítulo 4 - Diagrama de Sequência do Sistema	15
4.1 - Diagrama de Sequência do Sistema - Fluxo Principal	15
4.1.1 - Descrição do Diagrama de Sequência do Sistema - Fluxo Principal	16
4.2 - Diagrama de Sequência do Sistema – Loop Mensagem	17
4.2.1 - Descrição do Diagrama de Sequência do Sistema – Loop Mensagem	18
4.3 - Diagrama de Sequência do Sistema – Fluxo Alternativo 1 (Ligação)	18
4.3.1 - Descrição do Diagrama do Sistema – Fluxo Alternativo 1 (Ligação)	19
4.4 - Diagrama de Sequência do Sistema – Fluxo Alternativo 2 (Término do tempo limite)	20
4.4.1 - Descrição do Diagrama de Sequência do Sistema - Fluxo Alternativo 2 (Término do tempo limite)	20
4.5 - Diagrama de Sequência do Sistema – Fluxo de Exceção 1	20
4.5.1 - Descrição do Diagrama de Sequência do Sistema - Fluxo de Exceção 1	21
4.6 - Diagrama de Sequência do Sistema – Fluxo de Exceção 2	21
4.6.1 - Descrição do Diagrama de Sequência do Sistema – Fluxo de Exceção 2	21
4.7 - Diagrama de Sequência do Sistema – Fluxo de Exceção 3	21
4.7.1 - Descrição do Diagrama de Sequência do Sistema – Fluxo de Exceção 3	22
Capítulo 5 - Diagrama de Pacotes	22
5.1 - Diagrama de Pacotes	22
5.1.1 - Descrição do Diagrama de Pacotes	23
Capítulo 6 - Diagrama de Classes de Projeto	25
6.1 - Diagrama de Classes do Projeto	27
6.2 - Descrição do Diagrama de Classes do Projeto	28

Capítulo 7 - Diagrama de Sequência de Projeto	36
7.1 - Diagrama de Sequência Fluxo Principal	36
7.1.1 - Descrição do Diagrama de Sequência Fluxo Principal	37
7.2 - Diagrama de Sequência Loop Mensagem	38
7.2.1 - Descrição do Diagrama de Sequência Loop Mensagem	38
7.3 - Diagrama de Sequência Fluxo Alternativo 2	39
7.3.1 - Descrição do Diagrama de Sequência Fluxo Alternativo 2	39
Capítulo 8 - Diagrama de Atividades	40
8.1 - Diagrama de Atividades	41
8.1.1 - Descrição do Diagrama de Atividades	42
Capítulo 9 - Diagrama de Estados	44
9.1 - Diagrama de Estado	45
9.1.1 - Descrição do Diagrama de Estado	45
Capítulo 10 - Conclusão	47
Referências Bibliográficas	48
Apêndice A - Lista de Categorias	49
Apêndice B - Análise Gramatical	50

## Lista de Figuras

Figura 1 - Diagrama de Contexto	4
Figura 2 - Diagrama de Casos de Uso	5
Figura 3 - Descrição do Diagrama de Classes de Domínio	12
Figura 4 - Diagrama de Sequência do Sistema - Fluxo Principal	16
Figura 5 - Diagrama de Sequência do Sistema - Loop Mensagem	18
Figura 6 - Diagrama de Sequência do Sistema - Fluxo Alternativo 1 (Ligação)	19
Figura 7 - Diagrama de Sequência do Sistema - Fluxo Alternativo 2 (Término do tempo limite)	20
Figura 8 - Diagrama de Sequência do Sistema - Fluxo de Exceção 1	21
Figura 9 - Diagrama de Sequência do Sistema - Fluxo de Exceção 2	21
Figura 10 - Diagrama de Sequência do Sistema - Fluxo de Exceção 3	22
Figura 11 - Diagrama de Pacotes	23
Figura 12 - Diagrama de Classes do Projeto	28
Figura 13 - Diagrama de Sequência Fluxo Principal	37
Figura 14 - Diagrama de Sequência Loop Mensagem	38
Figura 15 - Diagrama de Sequência Fluxo Alternativo 2	39
Figura 16 - Diagrama de Atividades	42
Figura 17 - Diagrama de Estado	45

## Lista de Tabelas

Tabela 1 – Descrição Completa Caso de Uso Tirar Dúvidas	9
Tabela 2 – Listas de Categorias	49
Tabela 3 – Análise Gramatical	50

## **Resumo**

Este trabalho provê a comunicação de estrangeiros em diferentes línguas utilizando o aplicativo HelloUniverse com a implementação de uma extensão para este sistema. Esta extensão apresenta diversas opções para o turista para ajudá-lo a conviver melhor no destino escolhido por ele.

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 (LARMAN, Craig)<sup>1</sup>. Além disso, ela ajuda a compreender melhor o sistema que será desenvolvido.

---

<sup>1</sup> Utilizando UML e Padrões. 3ª Edição. 2005.

## **Capítulo I - Introdução**

Muitos estrangeiros, quando viajam para outros países, enfrentam dificuldades para se comunicarem em outras línguas. Neste capítulo, apresentaremos o aplicativo HelloUniverse com a extensão MyUniverse, que tem o propósito de facilitar a comunicação dos viajantes em diferentes regiões do país que estão visitando.

### **I.1 - HelloUniverse**

O aplicativo HelloUniverse tem como objetivo permitir a comunicação de uma pessoa em um país estrangeiro. Sua finalidade é ser um tradutor simultâneo, através de uma ligação para um intérprete e o mesmo faz a ponte entre pessoas que falam línguas diferentes como inglês, espanhol, francês, alemão ou italiano. O custo dessas ligações são créditos adquiridos através de um cartão pré-pago com o código de acesso ou da instalação do próprio aplicativo.

### **I.2 - MyUniverse**

Como extensão para o aplicativo HelloUniverse, apresentamos o MyUniverse, que será uma funcionalidade de guia turístico, que junto com os dados do cadastro anterior a extensão, o usuário escolhe um primeiro pacote do país/cidade/região do seu interesse. Cada pacote é um guia da localidade que escolheu, contendo telefones emergenciais (polícia, bombeiros, SAMU – ou respectivo do país em questão -, entre outros), pontos turísticos com todas as informações sobre eles (horários de funcionamento, preços, endereço, descrição), clima do dia e da semana, algumas gírias locais, dicionário com algumas palavras chave da língua, frases básicas para interação e a diferença de fuso horário entre a região do pacote e onde o usuário mora. A extensão mandará mensagens de texto e/ou de áudio para o usuário mostrando situações nas quais pode usar as frases de interação, ensinando o usuário a se comunicar no lugar que o pacote se refere. Esta função de mensagens pode ser desativada pelo usuário a qualquer momento.

O MyUniverse permite que o próprio usuário possa enviar uma mensagem para intérpretes cadastrados caso tenha alguma dúvida específica, o que não deixa o usuário dependente apenas das ligações para tirar dúvidas. Essa mensagem será entregue a todos os intérpretes disponíveis que saibam as duas línguas do usuário. Assim que o primeiro intérprete responder, a mensagem some para os outros intérpretes e o usuário terá uma hora para tirar suas dúvidas com o mesmo intérprete. As mensagens de resposta poderão ter duas partes, uma com o conteúdo na língua

materna e a outra na língua local, e estas mesmas mensagens podem ser salvas em um pacote que o usuário escolha (só tendo que ter sido comprado antes).

O pagamento destes serviços é feito através dos créditos adquiridos no próprio aplicativo ou através do cartão comprado separado. Somente o primeiro pacote escolhido ao se cadastrar será gratuito e os seguintes dependem da quantidade de informações que os mesmos terão, mas sempre serão comprados com créditos. Os créditos são utilizados para mandar mensagens com dúvidas específicas também.

À medida que é utilizado, o aplicativo traça o perfil do usuário com as suas dificuldades e manda sugestões para o aprendizado do mesmo, e conteúdos de interesse do usuário. Estes conteúdos podem ser escolhidos no cadastro ou durante a utilização do aplicativo, através dos pacotes adquiridos e do assunto das dúvidas, por exemplo. Para não utilizar muita memória do celular, as mensagens e pacotes salvos serão armazenados em uma nuvem, ocupando pouca memória do celular. Essa nuvem poderá ser acessada com a internet do celular da pessoa ou através dos créditos de internet, também comprados pelo aplicativo ou em cartões comprados em lojas. Ao se cadastrar, o usuário recebe 30 créditos de internet, cada crédito dando direito à uma hora de acesso a nuvem, podendo ser não sequencial.

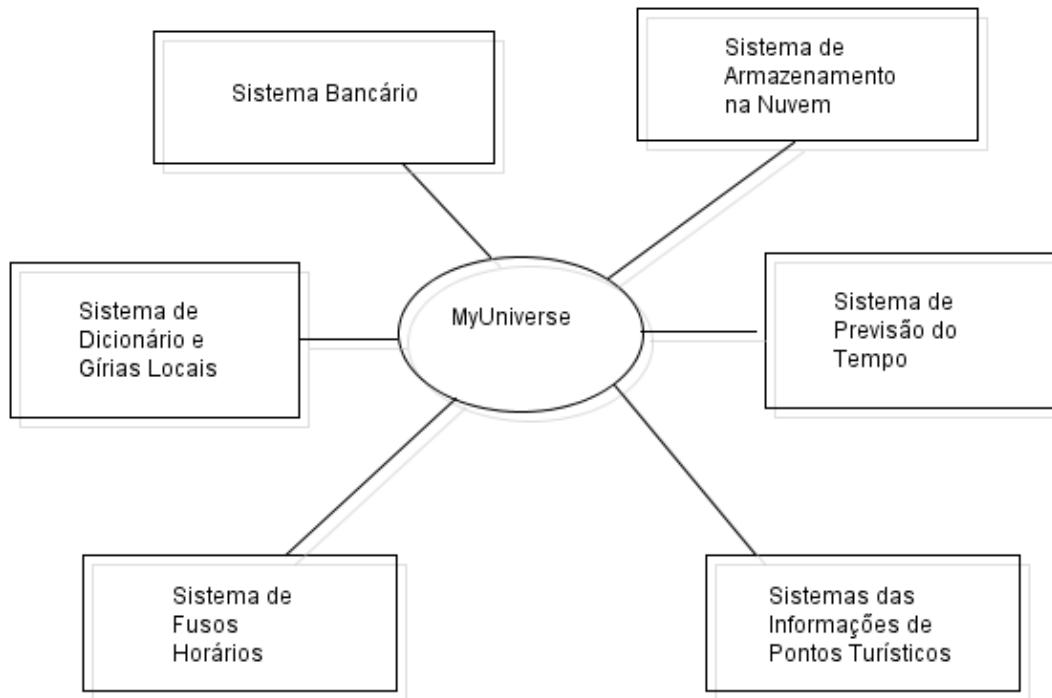
### **I.3 - Diagrama de Contexto**

O diagrama de contexto apresenta quais sistemas externos já existentes interagem com o sistema a ser construído. Este diagrama admite que o usuário tenha uma visão geral do escopo do problema.

No diagrama referente ao MyUniverse, temos 6 sistemas externos que se comunicarão com o nosso sistema: de armazenamento na nuvem, de previsão do tempo, de dicionário e gírias locais, de fusos horários e das informações de pontos turísticos. Vamos explicar cada um desses agora:

- Sistema de Armazenamento na Nuvem: para não ocupar muita memória no celular do usuário, as informações dos pacotes de cada região são armazenadas em um sistema externo.
- Sistema de Previsão do Tempo: as informações sobre o clima diário e da semana das regiões de cada pacote podem ser encontradas nesse sistema.
- Sistema de Dicionário e Gírias Locais: já existe um sistema que ensina a interagir em várias línguas diferentes e usaremos este sistema para essa parte dos pacotes.





**Figura 1 - Diagrama de Contexto**

- Sistema de Fusos Horários: sistema que dá os horários locais de cada região tirando daí a diferença do fuso horário da região e o local.
- Sistema das Informações de Pontos Turísticos: sistema que apresenta os pontos turísticos de cada região, junto com uma breve descrição, horário de funcionamento, local e preços das atrações.
- Sistema Bancário: sistema que verifica a validação da aquisição dos créditos para a compra dos pacotes na hora do cadastro ou quando o usuário quiser comprar mais pacotes.

## Capítulo II - Diagrama de Casos de Uso

O diagrama de casos de uso descreve as funcionalidades que um sistema a ser projetado representa na visão do usuário, auxiliando na comunicação entre o analista de sistema e o usuário e no levantamento dos requisitos do sistema. Este diagrama é composto por atores, conjunto de casos de uso e relacionamentos.

Os atores são entidades externas com quem o sistema em desenvolvimento interage. A partir da identificação dos atores, podemos construir os casos de uso, levando em consideração o que cada ator precisa que o sistema faça. Nessa construção, pode-se distinguir entre dois tipos de casos de uso: primários, que representam os objetivos básicos dos atores ao acessar o sistema e os secundários, que não trazem benefícios diretos para os atores, mas são necessários para que o sistema funcione adequadamente.

### II.1 - Diagrama de Casos de Uso do Sistema

Importante: a explicação e o uso do "includes" estão errados.

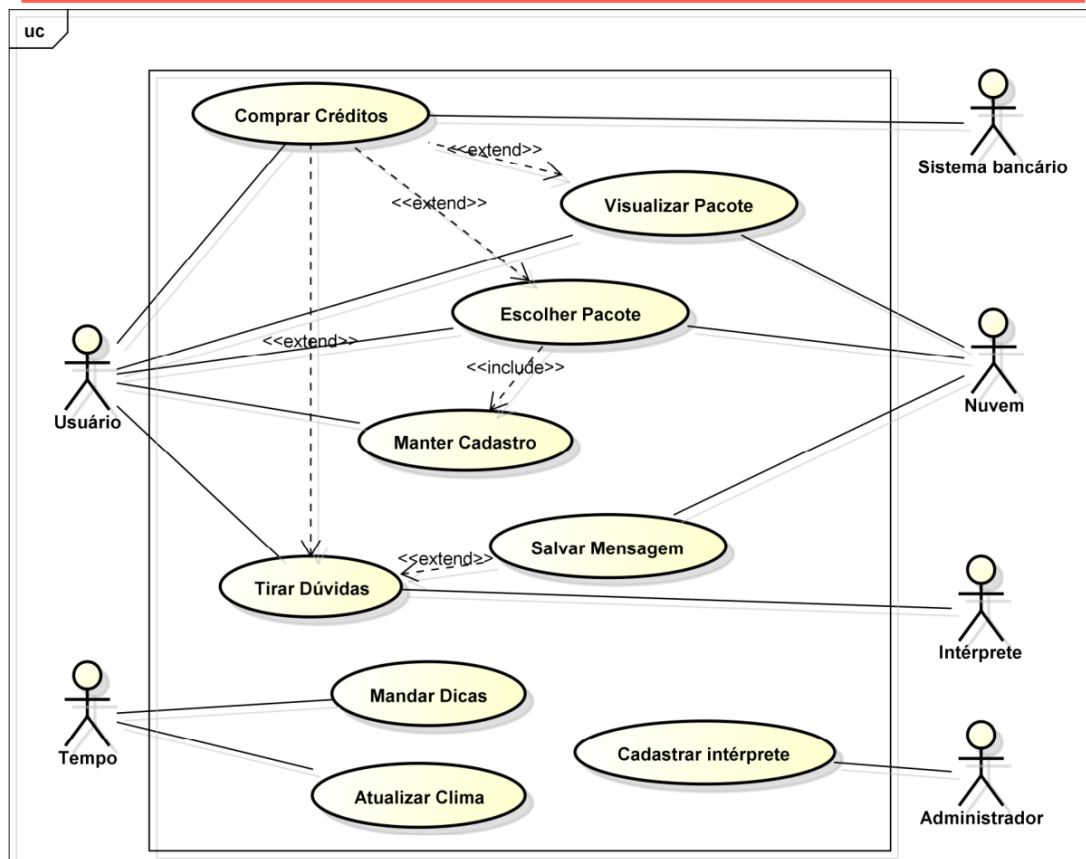


Figura 2 - Diagrama de Casos de Uso

### II.2 - Descrição do Diagrama de Casos de Uso

Em relação ao diagrama de casos de uso acima temos a seguinte configuração:

- Atores:

É representado por um boneco com o nome do ator. Neste caso, o Usuário, o Tempo, o Sistema bancário, a Nuvem, o Intérprete e o Administrador são os atores.

- 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, Comprar Créditos, Visualizar Pacote, Escolher Pacote, Manter Cadastro, Salvar Mensagem, Tirar Dúvidas, Mandar Dicas, Atualizar Clima e Cadastrar Intérpretes são os casos de uso.

- Relacionamento:

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:

- a) **Include:** Uma interação entre casos de uso que mostra uma relação de dependência entre eles. Por exemplo, a interação entre Escolher Pacote e Manter Cadastro, pois é necessário que o usuário já tenha feito seu cadastro para então poder escolher seu pacote de interesse e o caso de uso Manter Cadastro não está completo.
- b) **Extend:** Uma interação entre casos de uso que mostra uma relação de extensão entre eles. Por exemplo, a interação entre Comprar Créditos e Escolher Pacote que apresenta a ideia de que o usuário pode Comprar Créditos enquanto está escolhendo um pacote, mas Escolher Pacote é um caso de uso completo.
- c) **Associação:** A interação entre ator e caso de uso onde há troca de informações entre eles. Por exemplo, Usuário e Manter Cadastro.

## II.3 - Descrição dos Atores e Casos de Uso

### II.3.1 - Descrição dos atores

- Usuário:

O Usuário efetua ligações ou manda mensagens para os Intérpretes a fim de tirar dúvidas.

- Tempo:

O Tempo é ligado aos casos de uso temporais, que são acionados com certa frequência.

- Sistema bancário:

Atenção: explicação do includes errada.

O Sistema bancário é responsável pela administração de pagamentos de créditos realizados por Usuários.

- Nuvem:

A Nuvem é responsável pelo armazenamento de mensagens e gerenciamento de pacotes no sistema.

- Intérprete:

O Intérprete é responsável em ajudar o Usuário na comunicação de diferentes idiomas.

- Administrador:

O Administrador é responsável pelo gerenciamento de cadastros de Intérpretes.

### **II.3.2 - Descrição dos casos de uso**

A seguir apresentamos a descrição resumida de cada caso de uso do nosso diagrama:

- Comprar Créditos:

O usuário entra no aplicativo e escolhe a opção comprar créditos. Em seguida, seleciona se deseja créditos ou créditos de internet e a quantidade de créditos que deseja comprar. Depois, informa a sua forma de pagamento. Caso opte comprar créditos por meio de cartão pré-pago, basta procurar um local mais próximo para adquirir um cartão com o código de acesso e depois de inserido no aplicativo, confirmar a quantidade de créditos comprados. Caso o usuário prefira comprar créditos pelo cartão de crédito, o Sistema apresenta a tela para inserir o cartão e a senha e contacta o Sistema Bancário para validar a compra.

- Manter Cadastro:

O usuário entra no aplicativo e caso não tenha uma conta de acesso, o mesmo seleciona a opção realizar cadastro e informa todos os seus dados pessoais, além dos idiomas falados e alguns conteúdos de interesse. Também seleciona algumas configurações, como utilizar a rede do celular, móvel ou wi-fi, ou créditos de internet para acessar os pacotes. Em seguida, escolhe o pacote que deseja gratuitamente.

- Mandar Dicas:

O sistema envia, toda semana, sugestões para o usuário realizar durante o passeio, conteúdos do interesse do mesmo e gírias da região, tudo conforme o perfil traçado a partir das compras dos pacotes.

- Escolher Pacote:

O usuário entra no aplicativo e escolhe a opção 'escolher pacote'. Feito isso, o usuário pode escolher um pacote de acordo com a sua preferência. Apenas o primeiro

pacote escolhido ao se cadastrar é gratuito. Caso não seja o primeiro acesso e o usuário não tiver créditos suficientes, o sistema fará o caso de uso 'comprar créditos' e só depois o usuário poderá usufruir da quantidade de informações que comprou. E os pacotes são armazenados na nuvem para otimizar o espaço da memória do celular.

- Visualizar Pacote:

O usuário escolhe a opção 'visualizar pacotes'. O sistema mostra os pacotes comprados pelo usuário, este escolhe um pacote. Sendo assim, o sistema verifica se o usuário tem créditos para acessar a nuvem, caso o mesmo quis usá-los, informado no cadastro. Se ele tiver, o sistema mostra os seguintes dados ao acessar a nuvem: pontos turísticos, gírias locais, dicionário com algumas palavras chaves da língua, frases básicas para interação e a diferença de fuso horário entre a região do pacote e onde o usuário mora. O usuário escolhe uma opção e as informações são mostradas.

- Tirar Dúvidas:

O usuário escolhe a opção tirar dúvidas. O sistema pergunta de qual língua o usuário quer tirar dúvidas e mostra quais das línguas cadastradas como falantes pelo usuário tem intérprete que saiba falar as duas línguas (a falada e a em dúvida). Depois que o usuário escolher o duo de línguas que deseja, o sistema pergunta qual a forma que o usuário quer tirar as dúvidas, se ligando ou mandando mensagem. Enquanto o usuário escolhe o modo, o sistema verifica se o usuário tem créditos suficientes. Escolhendo ligar, o sistema faz a ligação para um dos intérpretes disponíveis e o usuário poderá tirar suas dúvidas. Quando acabar, o usuário desliga a ligação e confere os créditos usados e avalia a ligação. Caso escolha mandar mensagem, o sistema deixará o usuário escrevendo a mensagem enquanto avisa a todos os intérpretes disponíveis que tem um usuário querendo tirar dúvidas. Assim que um dos intérpretes aceitar a mensagem, o usuário terá até uma hora para tirar suas dúvidas. Quando o usuário quiser sair, só escolher sair que o sistema apresenta os créditos restantes na conta e uma avaliação do atendimento.

- Atualizar Clima:

Todo dia o aplicativo entra no sistema de previsão de tempo e pega as informações atualizadas do clima diário e semanal das regiões dos pacotes do usuário.

- Cadastrar Intérprete:

O administrador entra no sistema e seleciona a opção cadastrar intérprete. O sistema apresenta uma tela com os campos dos dados pessoais do intérprete para o administrador preencher. O mesmo seleciona os idiomas que o intérprete tem conhecimento. Para finalizar o cadastro, o sistema salva os dados preenchidos.

- Salvar Mensagem:

Durante a execução do caso de uso Tirar dúvidas, o usuário pode salvar as mensagens recebidas do intérprete. Caso queira, o usuário seleciona as mensagens e escolhe a opção salvar, e o sistema então mostra os pacotes onde o usuário poderá salvar as mensagens. Depois de selecionado o pacote, o sistema salva as mensagens na nuvem.

Agora apresentamos a descrição completa de um caso de uso escolhido do nosso diagrama:

**Tabela 1 – Descrição Completa Caso de Uso Tirar Dúvidas**

Nome	Tirar Dúvidas
Identificador	CSU01
Importância	Risco alto Prioridade alta
Sumário	O usuário usa o sistema para tirar dúvidas em uma língua estrangeira
Ator primário	Usuário
Ator secundário	Intérprete
Pré-condições	O usuário deve estar logado
Fluxo principal	<ol style="list-style-type: none"><li>1. Usuário entra na opção “tirar dúvidas”</li><li>2. Sistema pergunta de qual língua o usuário quer tirar dúvidas</li><li>3. Usuário escolhe a língua</li><li>4. Sistema verifica se o usuário tem créditos suficientes.</li><li>5. Sistema mostra qual das línguas faladas pelo usuário (citadas no cadastro) tem algum intérprete disponível.</li><li>6. Usuário escolhe o duo de línguas que deseja.</li><li>7. Sistema pergunta qual a forma que o usuário quer tirar as dúvidas, ligando ou mandando mensagem.</li><li>8. Usuário escolhe mandar mensagem.</li><li>9. Sistema avisa a todos os intérpretes disponíveis que tem um usuário querendo tirar dúvidas.</li><li>10. Um dos intérpretes aceita a mensagem.</li><li>11. Sistema cancela a notificação dos outros intérpretes.</li><li>12. Sistema gera um protocolo de atendimento.</li><li>13. Sistema permite a comunicação entre usuário e intérprete durante uma hora.</li><li>14. Usuário envia a mensagem.</li></ol>

	<p>15. Sistema repassa a mensagem para o intérprete.</p> <p>16. Intérprete responde a mensagem.</p> <p>17. Sistema apresenta a mensagem para o usuário.</p> <p>18. Se tiver mais dúvidas, retornar ao passo 14.</p> <p>19. Usuário sai a qualquer momento dentro dessa uma hora.</p> <p>20. Sistema avisa ao intérprete que acabou o atendimento.</p> <p>21. Sistema apresenta um questionário de avaliação do atendimento para o usuário.</p> <p>22. Usuário responde a avaliação.</p> <p>23. Sistema apresenta os créditos restantes na conta, o número do protocolo de atendimento e o caso de uso termina.</p>
Fluxo alternativo	<p>Fluxo alternativo 1 (8): Ligação</p> <p>a. Usuário escolhe ligar.</p> <p>b. Sistema liga para um dos intérpretes disponíveis.</p> <p>c. Caso o Intérprete não atenda, caso de uso volta para o passo anterior (fluxo alternativo 1 - passo b). Caso contrário, o intérprete atende.</p> <p>d. Sistema gera um protocolo de atendimento.</p> <p>e. Sistema libera a ligação para o usuário, permitindo a comunicação entre eles.</p> <p>f. Usuário se comunica com o intérprete.</p> <p>g. Sistema verifica a quantidade de créditos a cada minuto de ligação.</p> <p>h. Usuário desliga a ligação e o caso de uso retorna ao passo 21.</p> <p>Fluxo alternativo 2 (13): Término do tempo limite</p> <p>a. Caso termine a hora antes do usuário sair, sistema pergunta se o usuário deseja continuar a tirar dúvidas ou não.</p> <p>b. Caso o usuário aceite continuar, o sistema verifica os créditos em conta de novo e retorna ao passo 13. Caso o usuário queira acabar, o caso de uso retorna no passo 21.</p>
Fluxo de exceção	<p>Fluxo de exceção 1 (4): Violação da RN01 ou RN02.</p>

	<p>a. Se o usuário não tiver os créditos suficientes, o sistema avisa o usuário, termina esse caso de uso começando o “Comprar Créditos”.</p> <p>Fluxo de exceção 2 (FA1-f): Violação da RN01.</p> <p>a. Caso os créditos em conta tenham acabado, o sistema entra na ligação avisando o usuário e o intérprete automaticamente.</p> <p>b. Caso de uso retorna ao passo 21.</p> <p>Fluxo de exceção 3 (FA2-b): Violação da RN02.</p> <p>a. Caso os créditos em conta tenham acabado, o sistema avisa ao usuário e o caso de uso retorna ao passo 20.</p>
Pós-condições	<p>O usuário tem toda(s) a(s) sua(s) dúvida(s) sanada(s), tem parte das suas dúvidas sanadas ou é redirecionado ao caso de uso “Comprar Créditos”.</p>
Regras de negócio	<p>Créditos para ligação (RN01) - O usuário precisa ter um crédito para falar um minuto com o intérprete.</p> <p>Créditos para mensagem (RN02) - O usuário precisa ter um crédito para mandar mensagens durante uma hora.</p>



## Capítulo III - Diagrama de Classes de Domínio

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 (o relacionamento entre os objetos das classes), podemos construir as classes considerando as respectivas identificações e, em seguida, descrever o modelo textual desse diagrama de classes.

### III.1 - Diagrama de Classes de Domínio do Sistema

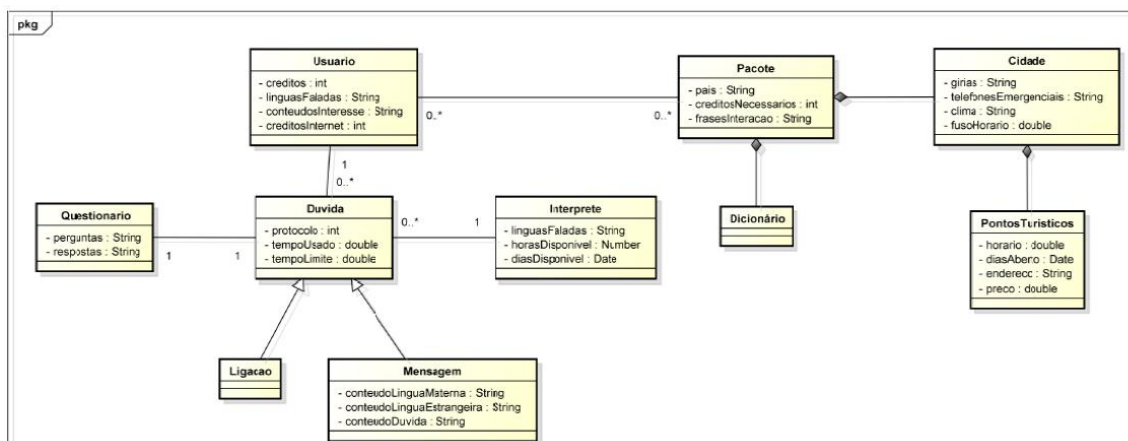


Figura 3 - Descrição do Diagrama de Classes de Domínio

#### III.1.1 - Descrição do Diagrama de Classes de Domínio

Partindo da técnica de análise gramatical (Apêndice B), que identifica possíveis classes, pegamos os casos de uso e destacamos os nomes ou pares de nome-adjetivo. A partir disso, organizamos cada palavra, com a ajuda de uma tabela e as classificamos em classe ou não. Também utilizamos outra técnica chamada lista de categorias (Apêndice A), para encontrarmos o máximo possível de classes. Com isso, tivemos base para construir o diagrama de classes, que será descrito nos próximos parágrafos.

A classe Usuario terá como alguns atributos: créditos, línguas faladas, conteúdos de interesse e créditos de internet. Estes terão seus dados armazenados privadamente. Além dos atributos, a classe terá relacionamentos, que são: um usuário terá de zero a vários pacotes e de zero a várias dúvidas.

A classe Pacote contém os seguintes atributos, que serão privados: pais, créditos necessários, frases para interação, entre outros. Nessa classe, temos casos especiais de associações, a relação todo-parte de composição com as classes Cidade e Dicionario, que se destruímos o todo (Pacote), implicaremos na destruição das partes (Cidade e Dicionário). Uma classe Pacote terá de zero a vários usuários associados.

A classe Cidade agrupa os seguintes atributos: gírias, telefones emergenciais, clima, fuso horario, entre outros. Esses dados serão armazenados como privados. Também terá relacionamentos todo-parte de composição, com as classes Pacote sendo a parte, e Ponto Turistico sendo o todo.

A classe Ponto Turistico terá como atributos, que serão dados armazenados como privados, os itens a seguir: horário, dias abertos, endereço, preco, entre outros. Essa classe também se relacionará como a parte da composição com a classe Cidade.

A classe Dicionario tem um relacionamento todo-parte com a classe Pacote sendo a parte.

A classe Duvida contém os seguintes atributos: protocolo, tempo usado, tempo limite, entre outros. Essas informações serão armazenadas como privadas. Essa classe é uma generalização das classes Ligação e Mensagem. Também poderá ter um único Interpretre e um único Usuario ligado a essa classe.

A classe Ligacao é uma especialização da classe Duvida e não tem atributos por não fazer parte da extensão.

A classe Mensagem terá como atributos de âmbito privado, os seguintes dados: conteúdo da duvida, conteúdo da língua materna, conteúdo da lingua estrangeira, entre outros. Essa classe será uma especialização da classe Duvida.

A classe Interpretre agrupa os seguintes atributos: línguas faladas, horas disponível, dias disponível, entre outros. Estes dados serão armazenados como privados. Essa classe se relacionará com a classe Duvida, sendo que cada intérprete poderá ter de zero a várias dúvidas para resolver.

A classe Questionario tem os seguintes atributos: perguntas e respostas, que serão armazenados como privados. Essa classe se relacionará com a classe Duvida do seguinte modo: um questionário só pode ter associado uma única dúvida e uma dúvida só pode ter um único questionário associado.

## **Capítulo IV - Diagrama de Sequência do Sistema**

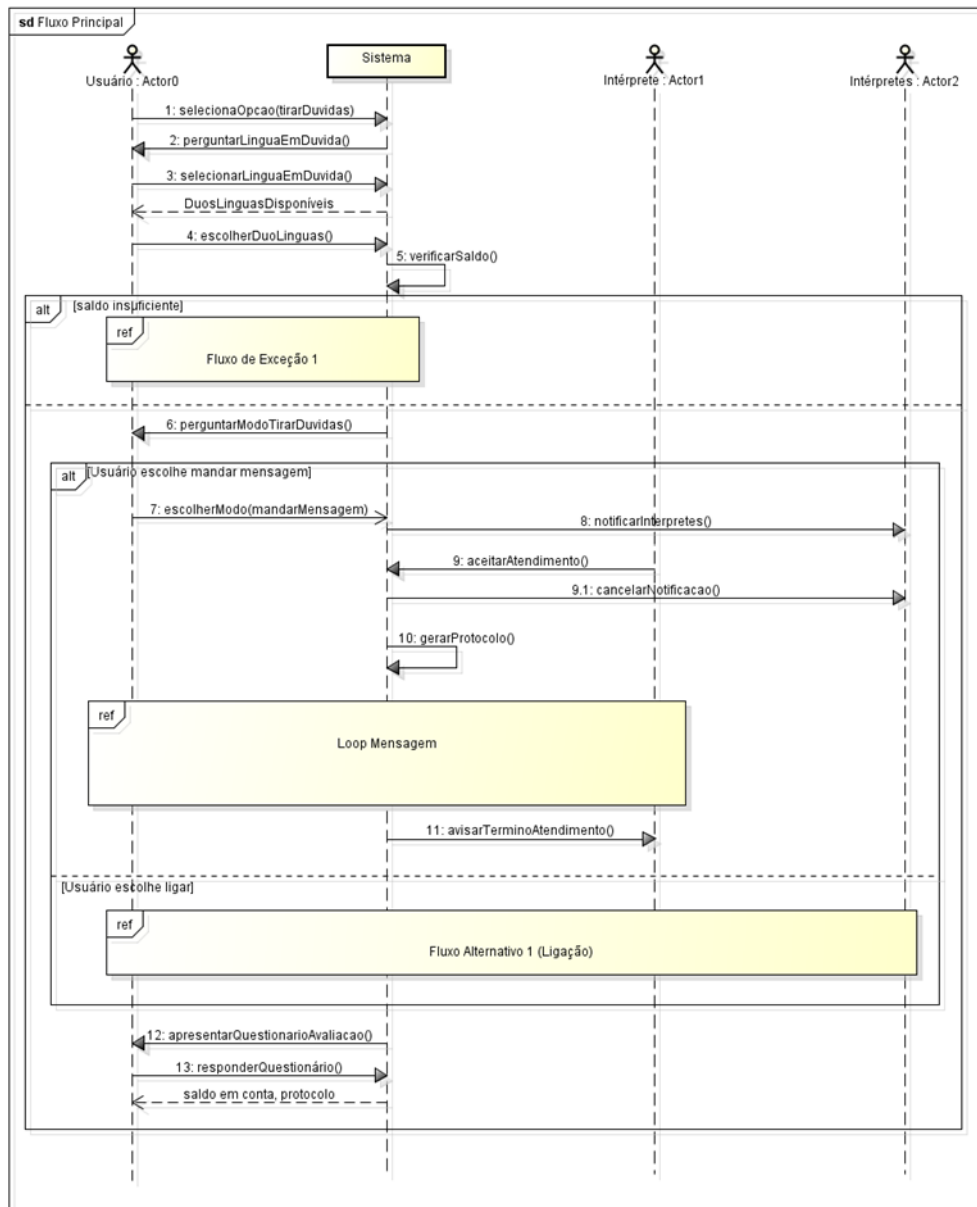
O diagrama de sequência do sistema é criado a fim de representar o comportamento dos objetos por meio de cenários de casos de uso. Neste capítulo, apresentaremos os diagramas de sequência para o caso de uso estendido apresentado no capítulo 2.

Um diagrama de sequência do sistema mostra os eventos para um fluxo de caso de uso específico a partir da interação entre os objetos. Os eventos são representados através de mensagens trocadas entre eles, podendo também ser enviada para o mesmo objeto, para solicitar que tal evento seja executado. A descrição do fluxo é iniciada com a mensagem enviada por um objeto, podendo ser ele ator primário ou sistema, dando início também a interação entre eles.

No diagrama, cada objeto é demonstrado no eixo vertical com o seu nome descrito na parte superior e por uma linha tracejada chamada de linha de vida, onde mostra o período de execução de cada evento que foi solicitado. As mensagens são representadas no eixo horizontal através de uma seta e são chamadas de síncronas (setas fechadas e preenchidas), quando o remetente aguarda a resposta antes de dar continuidade; assíncronas (setas abertas), quando o remetente não espera pela resposta para dar continuidade; de retorno (setas com linhas tracejadas), que especifica o resultado da mensagem enviada anteriormente; ou reflexiva, que especifica o envio de uma mensagem ao próprio remetente. Tais tipos são identificados com o nome e o argumento do evento apresentado, e seguem uma sequência lógica de acordo com a ordem cronológica do cenário de caso de uso em questão.

Também temos nos diagramas os quadros de interação, como: o sd, que encapsula o diagrama de sequência; o ref, que referencia um diagrama representado separadamente; o alt, que representa o “se-então-senão”, permitindo a definição de uma lógica no diagrama; o opt, que modela o “se-então”; e o loop, usado quando uma operação deve ser realizada várias vezes.

### **IV.1 - Diagrama de Sequência do Sistema - Fluxo Principal**



**Figura 4 - Diagrama de Sequência do Sistema - Fluxo Principal**

#### **IV.1.1 - Descrição do Diagrama de Sequência do Sistema - Fluxo Principal**

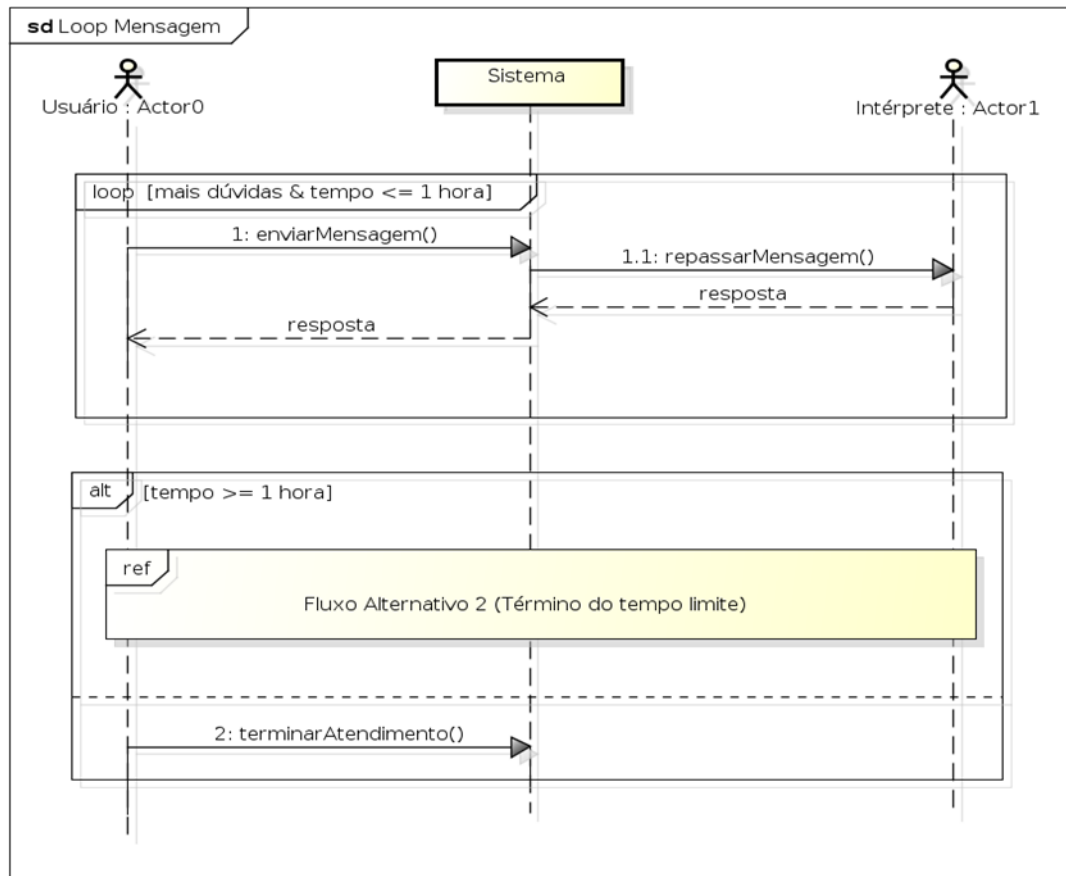
O diagrama de sequência ilustrado na figura 4 representa o Fluxo Principal do caso de uso estendido Tirar Dúvidas. Para começar, o Usuário seleciona a opção tirar dúvidas, mensagem do Usuário para o Sistema. Este pergunta ao Usuário qual a língua que deseja utilizar para tirar as dúvidas. Assim, o Usuário envia a opção da língua selecionada para o Sistema e este retorna a mensagem solicitada mostrando os duos de línguas que estão disponíveis para tirar as suas dúvidas. Em seguida, o Usuário escolhe qual duo quer e o Sistema verifica a quantidade de saldo em conta do Usuário.

Um novo fragmento encapsulado do tipo alt é criado a fim de que decisões sejam tomadas durante o fluxo. Nele, caso o Usuário não tenha saldo suficiente, o fluxo é direcionado diretamente para o diagrama Fluxo de Exceção 1, que será definido posteriormente, para poder comprar créditos. Caso contrário, o Sistema pergunta qual o modo que o Usuário deseja tirar as dúvidas, por meio de mensagem ou ligação.

Novamente, um novo fragmento encapsulado do tipo alt é criado, e se o Usuário escolher mandar mensagem, o Sistema notifica a todos os Intérpretes disponíveis que um Usuário necessita de ajuda. E quando o primeiro aceitar o atendimento, o Sistema imediatamente cancela a notificação enviada aos outros Intérpretes. O protocolo de atendimento será gerado. Posteriormente, no momento em que a comunicação entre o Usuário e o Intérprete é realizada, o fluxo é direcionado para o diagrama Loop Mensagem, definido separadamente.

Depois que o Loop Mensagem termina, o Sistema avisa ao Intérprete sobre o fim do atendimento com o Usuário. No entanto, caso o Usuário escolha ligar, o fluxo será direcionado para o diagrama Fluxo Alternativo 1 (Ligação), definido separadamente. Finalizado o atendimento, o Sistema apresenta ao Usuário um questionário de avaliação correspondente ao atendimento feito ao Intérprete, identificando se este foi bom, regular ou ruim, sendo respondido em seguida. Finalmente, o Sistema envia para o Usuário o saldo restante na conta e o protocolo de atendimento gerado anteriormente.

#### **IV.2 - Diagrama de Sequência do Sistema – Loop Mensagem**

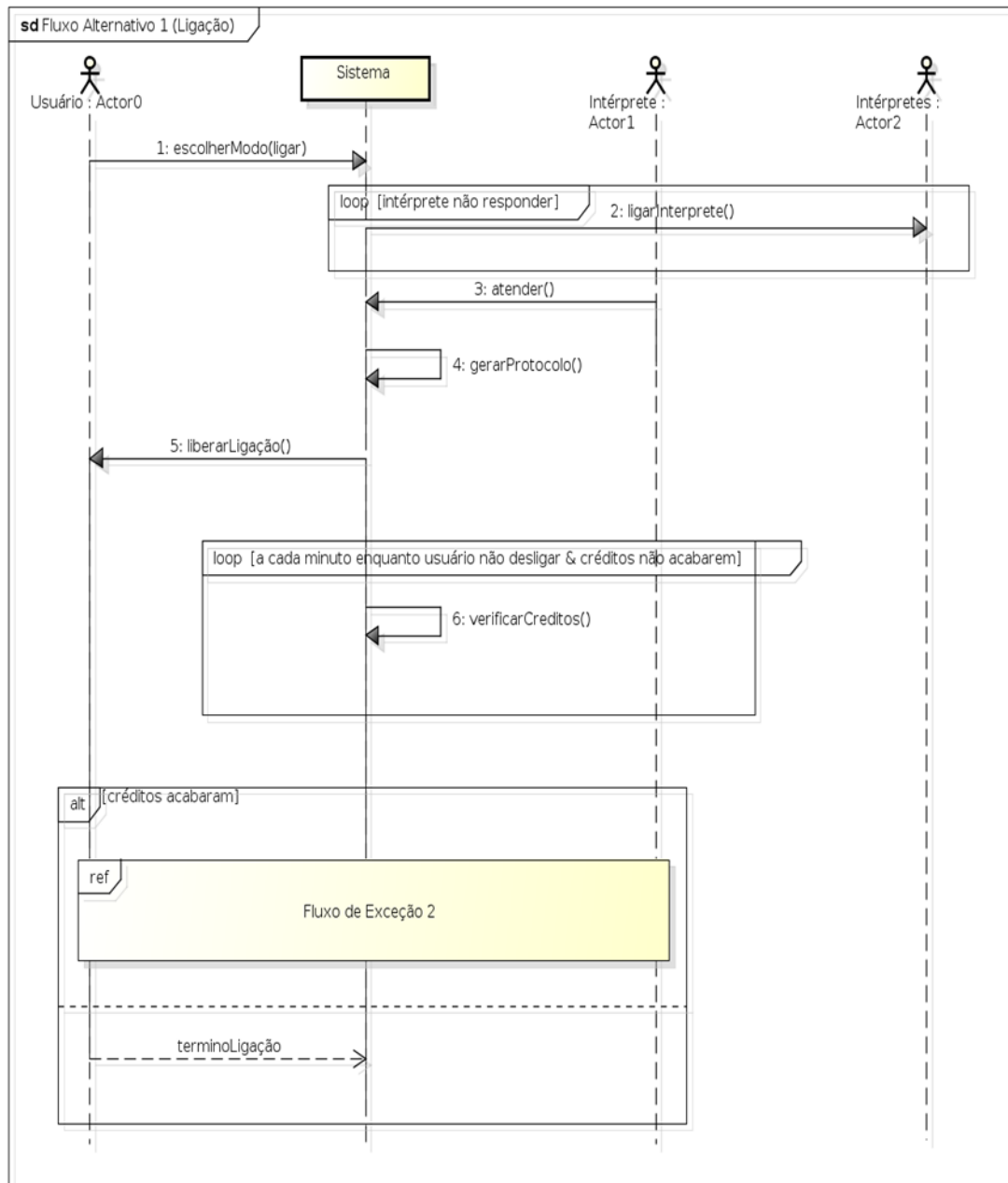


**Figura 5 - Diagrama de Sequência do Sistema - Loop Mensagem**

#### **IV.2.1 - Descrição do Diagrama de Sequência do Sistema – Loop Mensagem**

O diagrama de sequência ilustrado na figura 5 é referenciado no diagrama Fluxo Principal e no diagrama Fluxo Alternativo 2 (Término do tempo limite), e se inicia quando o Usuário envia mensagem para o Sistema com as dúvidas e este repassa para o Intérprete. Em seguida, o Intérprete retorna com uma resposta para o Sistema e este repassa para o Usuário. Estas interações se repetem enquanto o Usuário tiver mais dúvidas e o tempo de conversa entre ele e o Intérprete ainda for menor ou igual a 1 hora. Posteriormente, um novo fragmento encapsulado do tipo alt é criado e se tempo de conversa for maior que 1 hora, o fluxo é redirecionado para o diagrama Fluxo Alternativo 2 (Término do tempo limite). Caso contrário, significa que o Usuário não tem mais dúvidas então envia uma mensagem para o Sistema informando o término do atendimento.

#### **IV.3 - Diagrama de Sequência do Sistema – Fluxo Alternativo 1 (Ligação)**



**Figura 6 - Diagrama de Sequência do Sistema - Fluxo Alternativo 1 (Ligação)**

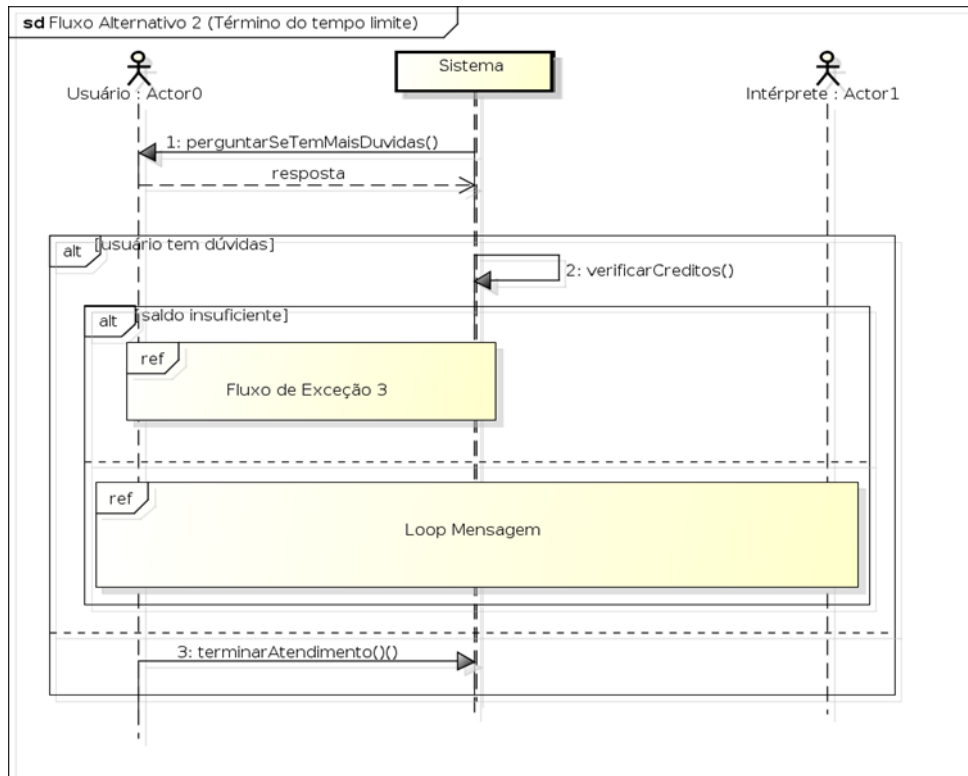
#### **IV.3.1 - Descrição do Diagrama do Sistema – Fluxo Alternativo 1 (Ligação)**

O diagrama de sequência ilustrado na figura 6 é referenciado no diagrama Fluxo Principal e se inicia quando o Usuário escolhe como modo de tirar dúvidas, a ligação. Neste momento, o Sistema liga para um dos Intérpretes e caso não atenda, o evento se repetirá até que um deles atenda a ligação. Quando um Intérprete atende, este avisa ao Sistema e um protocolo de atendimento é gerado. Logo, o Sistema libera a ligação para o Usuário. Durante a conversa entre o Usuário e o Intérprete. Enquanto o primeiro não encerre a ligação e os seus créditos não acabarem, a cada minuto é verificado a quantidade de créditos. Um novo fragmento encapsulado do tipo alt é criado, e se os créditos tiverem acabado, o fluxo é direcionado para o diagrama Fluxo



de Exceção 2, definido separadamente. Caso os créditos não tenham acabado, significa que o Usuário sanou todas as suas dúvidas, então o Usuário retorna com o término da ligação para o Sistema.

#### IV.4 - Diagrama de Sequência do Sistema – Fluxo Alternativo 2 (Término do tempo limite)



**Figura 7 - Diagrama de Sequência do Sistema - Fluxo Alternativo 2 (Término do tempo limite)**

##### IV.4.1 - Descrição do Diagrama de Sequência do Sistema - Fluxo Alternativo 2 (Término do tempo limite)

O diagrama de sequência ilustrado na figura 7 é referenciado no diagrama Loop Mensagem e este se inicia quando o tempo de 1 hora para conversar com o Intérprete por mensagem termina e o Sistema pergunta ao Usuário se ele tem mais dúvidas. Então, um novo fragmento encapsulado do tipo alt é criado. Caso o Usuário tenha mais dúvidas, o Sistema verifica se o Usuário tem créditos e outro fragmento encapsulado do tipo alt é criado referente ao saldo insuficiente. Com o término dos créditos, o fluxo é redirecionado para o diagrama Fluxo de Exceção 3. Caso ainda tenha créditos, o fluxo é redirecionado para o diagrama Loop Mensagem. No entanto, se o Usuário não tiver dúvidas, este envia uma mensagem para o Sistema informando o término do atendimento.

#### IV.5 - Diagrama de Sequência do Sistema – Fluxo de Exceção 1

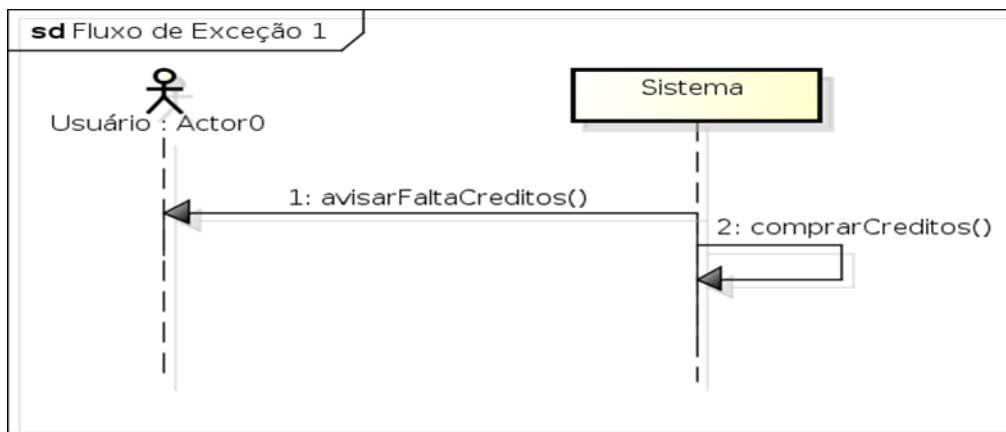


Figura 8 - Diagrama de Sequência do Sistema - Fluxo de Exceção 1

#### IV.5.1 - Descrição do Diagrama de Sequência do Sistema - Fluxo de Exceção 1

O diagrama de sequência ilustrado na figura 8 é referenciado no diagrama Fluxo Principal e representa uma exceção em que antes de o Usuário escolher o modo de tirar dúvidas com o Intérprete, o Sistema verifica que o mesmo não contém créditos suficientes para seguir adiante e o avisa sobre a falta desses créditos e em seguida, começa o caso de uso Comprar créditos, representado pela mensagem reflexiva.

#### IV.6 - Diagrama de Sequência do Sistema – Fluxo de Exceção 2

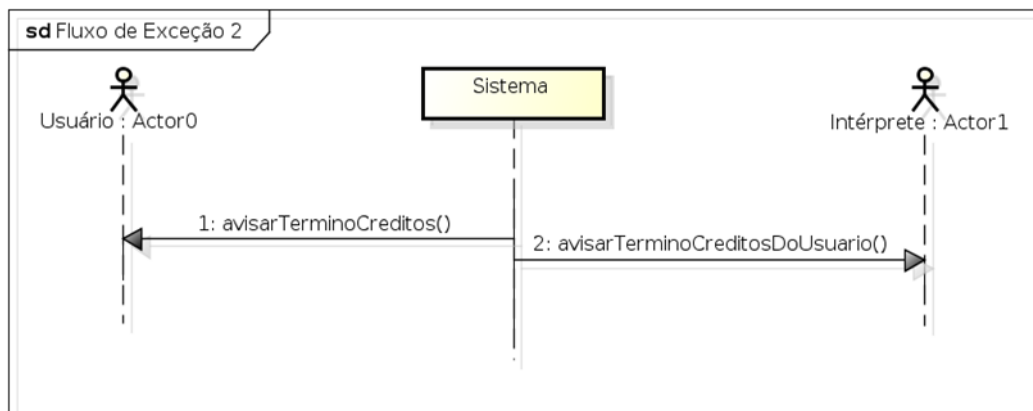


Figura 9 - Diagrama de Sequência do Sistema – Fluxo de Exceção 2

#### IV.6.1 - Descrição do Diagrama de Sequência do Sistema – Fluxo de Exceção 2

O diagrama de sequência ilustrado na figura 9 é referenciado no diagrama Fluxo Alternativo 1 (Ligação) e representa uma exceção caso os créditos do Usuário acabarem durante uma ligação do mesmo com um Intérprete. Assim, o Sistema avisa ao Usuário sobre o término dos créditos e ao mesmo tempo, avisa ao Intérprete que os créditos do Usuário com quem estava conversando acabaram.

#### IV.7 - Diagrama de Sequência do Sistema – Fluxo de Exceção 3



**Figura 10 - Diagrama de Sequência do Sistema – Fluxo de Exceção 3**

#### **IV.7.1 - Descrição do Diagrama de Sequência do Sistema – Fluxo de Exceção 3**

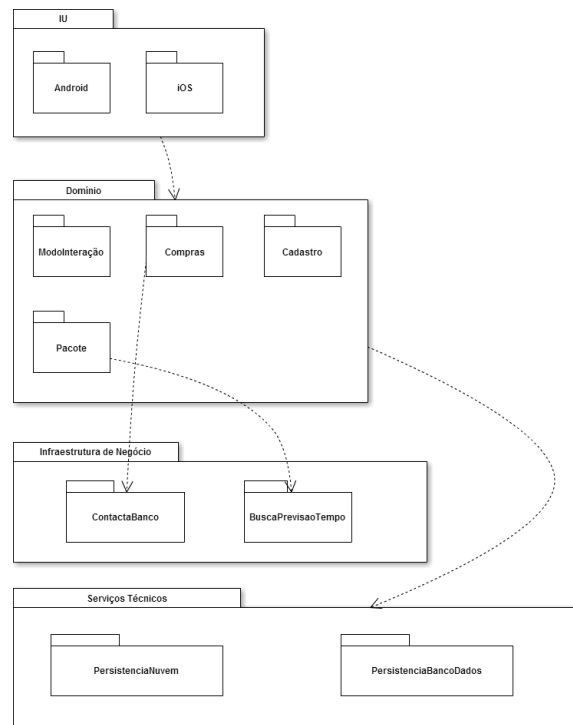
O diagrama de sequência ilustrado na figura 10 é referenciado no diagrama Fluxo Alternativo 2 (Término do tempo limite) e representa uma exceção caso o Usuário não tenha mais créditos para continuar a tirar dúvidas e o mesmo é avisado pelo Sistema sobre o término.

### **Capítulo V - Diagrama de Pacotes**

O Diagrama de Pacotes apresenta a descrição dos pacotes ou dos pedaços do sistema divididos em agrupamentos lógicos mostrando as dependências entre eles. Este diagrama é muito utilizado para ilustrar a arquitetura lógica de um sistema mostrando o agrupamento de suas classes. Um pacote representa um grupo de classes. Os pacotes se relacionam com outros pacotes através de uma relação de dependência. Um diagrama de pacotes pode ser utilizado em qualquer fase do processo de modelagem e visa organizar os modelos. Sua representação é uma pasta com o nome no interior ou na pega.

No diagrama, podemos ver que ele é representado por fatias verticais que são chamadas de camadas, onde cada camada contém um agrupamento de classes ou pacotes, que são representadas em uma divisão horizontal, conhecida como partições horizontais. Ele é dividido em camadas de Interface com Usuário (IU), Aplicação, Domínio, Infraestrutura do Negócio e Serviços Técnicos.

#### **V.1 - Diagrama de Pacotes**



**Figura 11 - Diagrama de Pacotes**

#### **V.1.1 - Descrição do Diagrama de Pacotes**

Em nosso diagrama temos como Interface com o Usuário (IU) as interfaces do sistema para Android e para iOS, que apresentam a forma como o Usuário irá interagir com o aplicativo.

No Domínio temos todos os serviços que são essenciais para o aplicativo que são o ModoInteracao, que determina a forma como o Usuário irá se comunicar com o Intérprete, por ligação ou por mensagem, mediados pelo Sistema; Compras, que engloba o que o usuário pode comprar, Créditos ou Pacotes; Cadastro, onde são fornecidas as informações do Usuário e do Intérprete, seus dados pessoais, além dos idiomas falados; Pacote, são todas as informações de todos os pacotes que o usuário já escolheu, contendo o país, frases de Interação e o Dicionário; e Cidade, apresenta todas as informações sobre os Pontos Turísticos, Gírias, Telefones Emergenciais, Clima e Fuso Horário. A IU é totalmente dependente do Domínio, pois ela é só uma interface com o Sistema.

Na Infraestrutura de Negócio temos o ContactaBanco, necessário para finalizar uma compra de créditos (que estarão no pacote Compras) e BuscaPrevisaoTempo, que tem como propósito atualizar o Clima do pacote Pacote, buscando essa informação no Sistema de Previsão do Tempo. É notório que Compras e Pacote, que fazem parte da camada de Domínio são dependentes do ContactaBanco e BuscaPrevisaoTempo, respectivamente, para realizar suas atividades.

Já nos Serviços Técnicos temos a PersistenciaNuvem, é o local onde ficarão salvas todas as informações a respeito do Usuário e seus pacotes, a partir da compra do primeiro; e PersistenciaBancoDados, onde serão armazenadas todas as informações de todos os pacotes, dos dicionários, e dos pontos turísticos, que vão ser sempre atualizadas 3 vezes por ano (início, meio e final do ano). Isso só será possível se todo o Domínio for dependente dos Serviços Técnicos, pois só assim que tudo poderá ser acessado e guardado. Os pacotes acabam armazenados nos dois Serviços Técnicos, no Banco de Dados, desde sua criação, e na Nuvem a partir da compra pelo Usuário.

## Capítulo VI - Diagrama de Classes de Projeto

O diagrama de classes de projeto é o modelo que consiste em um conjunto de representações de classes resultado das transformações ocorridas nas classes do diagrama de classes de domínio, a fim de torná-las mais detalhadas a partir de conceitos importantes aplicadas à elas, como os atributos, as operações, as associações, a herança e o polimorfismo. Este diagrama está muito próximo de um modelo de implementação.

Uma classe pode ser subdividida em três compartimentos, sendo o primeiro representado pelo nome que a identifica, o segundo pelo seus atributos e o terceiro pelas suas operações, chamada também de métodos.

No diagrama de classes de domínio, os atributos são representados apenas pelo nome, o que difere deste diagrama, pois apresenta a sua definição mais detalhada. Neste caso, primeiro declaramos a sua visibilidade. A visibilidade permite definir quais atributos de um objeto são acessíveis por outros objetos (BEZERRA, Eduardo)<sup>2</sup>. Eles podem ser públicos, privados, protegidos ou de pacote. O público é visível a objetos externos; o privado é visível apenas à objetos da própria classe; o protegido é visível às subclasses da classe e o de pacote é visível a classes do mesmo pacote. Este conceito serve para encapsular as estruturas internas da classe. Em seguida, vem o nome e o tipo de atributo. Este último podendo ser dos tipos abstratos de dados da linguagem e do programador ou dos tipos primitivos. Podemos ter também atributos derivados, cujo valor é calculado de outros atributos; estáticos, cujos valores são comuns a todos os objetos das classes.

Assim como os atributos, as operações, chamadas de métodos, neste diagrama também são mais elaboradas. É preciso que todos os métodos necessários relativos a parte do projeto em questão sejam identificados e adicionados ao diagrama. Como nos atributos, primeiro declaramos a sua visibilidade. Em seguida, vem o nome do mesmo, que é atribuído a partir de um verbo seguido de um complemento. Depois vem os parâmetros, que são as informações que o método recebe quando é executado. Cada método pode ter zero ou mais parâmetros, porém, quanto menor o número de parâmetros, menor é a dependência entre as classes. Depois dos parâmetros separados por vírgula, podemos ter em seguida o tipo de retorno, ou seja, o valor que o método retorna. Mas este caso depende da linguagem de programação que está sendo utilizada. Nem todas as operações retornam valor. E existem também os métodos estáticos, que são métodos da classe e podem ser

---

<sup>2</sup> Princípios de Análise e Projeto de Sistemas com UML. Capítulo 8.

chamados sem que a classe precise ser instanciada, o que é obrigatório para os métodos não estáticos.

São utilizadas também operações que interagem entre objetos, sendo elas de criação, que instancia um objeto, cujo método contém diferentes parâmetros, assim como o estado inicial do objeto a ser declarado, e de destruição, que libera memória alocada na instanciação, podendo ser aplicada diretamente pela linguagem. Além disso, são definidas operações para obter acesso e modificar cada atributo. São conhecidos como seletores, que retorna o valor de um atributo e modificadores, que altera o valor de um atributo.

O relacionamento associativo aplicado no diagrama de classes de domínio é definida como bidirecional, onde os objetos associados entre si já se conhecem. A fim de diminuir o acoplamento, o relacionamento associativo aplicado no diagrama deste capítulo é transformado para unidirecional, sempre que possível, onde a classe apontada não conhece a classe que aponta. É também aplicada neste diagrama o relacionamento de dependência em que uma classe depende dos serviços fornecidos por uma outra classe. O relacionamento entre duas classes, representada por uma linha direcionada e tracejada, podem ter dependência por variável global, por variável de parâmetro, por variável local e por método estático.

Pode ser aplicado também o relacionamento por herança, onde uma classe filha herda características da sua classe pai. A herança é representada em classes abstratas. A classe abstrata não pode ser instanciada e serve apenas para derivar outras classes. Nessa classe, o nome e o método abstrato, podendo ter mais de um e que não é implementado, são representados em itálico. A implementação do método abstrato da superclasse (classe pai) geralmente é feita nas subclasses (classes filhas), chamadas de classes concretas, e quando não é feita, essas subclasses também serão classes abstratas. A herança também é representada em operações polimórficas. Este caso ocorre quando uma operação implementada em uma superclasse é implementada na subclasse com a mesma assinatura. A operação na subclasse pode responder a mesma mensagem da operação da superclasse, ou caso contrário, a subclasse pode conter a mesma assinatura da operação da superclasse, porém, reimplementado essa operação.

O relacionamento de uma classe pode ser feita com uma interface. Na interface existe apenas a assinatura de métodos. As classes que a utilizará será responsável por realizar a implementação das assinaturas, aplicando o

comportamento aos métodos e essas classes podem implementá-las de maneiras diferentes.

Ao construirmos um diagrama de classes do projeto, assim como o diagrama de sequência do projeto no próximo capítulo, temos que ver se as mensagens que um objeto recebe está de acordo com as responsabilidades que ele executa, sendo elas a de conhecimento, sobre objetos relacionados, por exemplo, e de realização, como criar um objeto. Neste caso, foi aplicado os conceitos de padrões GRASP, a fim de compreender as atribuições de responsabilidades a objetos.

Dos diferentes padrões GRASP conhecidos, foram aplicados quatro padrões básicos no diagrama. São eles Criador, Acoplamento Baixo, Controlador e Especialista.

O Criador é utilizado para determinar a classe que ficará responsável pela criação de uma nova classe, ou seja, aquela classe que agrega instâncias da nova classe a ser criada.

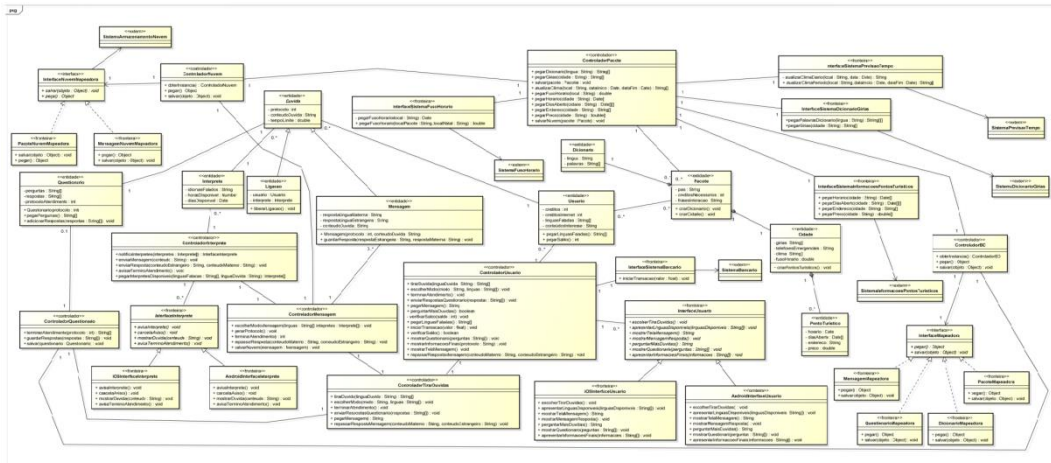
O Acoplamento Baixo é utilizado para atribuir responsabilidades de forma a minimizar o acoplamento, ou seja, é preciso encontrar uma classe que assuma determinada responsabilidade sem que seja muito dependente da outra.

O Controlador é utilizado para determinar a classe que será responsável por tratar eventos do sistema, sendo que esta classe não é de interface GUI, ou seja, é aquela classe que representa um tratador de eventos de sistema de um caso de uso (controlador caso de uso) ou aquela classe que representa o sistema todo (controlador fachada).

O Especialista é utilizado para atribuir responsabilidades, ou seja, é preciso determinar uma classe que contenha as informações necessárias para satisfazer a responsabilidade atribuída.

## **VI.1 - Diagrama de Classes do Projeto**





**Figura 12 - Diagrama de Classes do Projeto**

## VI.2 - Descrição do Diagrama de Classes do Projeto

A descrição começará pelas classes de entidade, depois as classes de fronteira e por último as classes de controle. Nenhuma classe pode ser dita como completa, pois esse diagrama foi baseado em um único caso de uso, o Tirar Duvidas. Todos os métodos e atributos necessários para o caso de uso em questão estão identificados e mais alguns outros.

As classes de entidade identificadas são Pacote, Dicionario, Cidade, PontoTuristico, Usuario, Interpretre, Questionario, Duvida, Ligacao e Mensagem. A seguir a descrição de cada uma delas:

- **Pacote:** classe que contém informações sobre cada pacote que o usuário já tenha comprado. Seus atributos são: `pais`, que guarda o país ao qual o pacote faz referência; `creditosNecessarios`, que guarda a quantidade de créditos necessária para comprar o pacote em questão e `frasesInteracao` que guarda algumas frases para interação no país do pacote. Seus métodos são `criarDicionario()` e `criarCidade()`, que criam objetos das duas classes que se relacionam por composição com `Pacote`, isto é, `Dicionario` e `Cidade` são parte de `Pacote` e se `Pacote` for destruído, as outras também serão. `Pacote` é composto de uma `Cidade` e de um `Dicionario`.
- **Dicionario:** classe que contém as informações de dicionário. Seus atributos modelados foram: `lingua`, que armazena a língua na qual o dicionário foi escrito e `palavras`, que guarda as palavras do dicionário com seus significados.
- **Cidade:** classe que guarda as informações referentes a uma cidade. Seus atributos identificados são: `gurias`, que apresenta as gírias locais; `telefonosEmergenciais`, onde são armazenados os telefones úteis do local, como o da polícia e o dos bombeiros; `clima`, que guarda as informações da

semana sobre o tempo na cidade, e fusoHorario, que apresenta a diferença de fuso horário entre a cidade do pacote e o endereço do usuário. Cidade é a parte do relacionamento todo-parte de composição com a classe Pacote e o todo no relacionamento todo-parte de composição com PontoTuristico.

- PontoTuristico: classe que guarda as informações sobre os pontos turísticos. Seus atributos são: horario, que é o horário de funcionamento do lugar; diasAberto, que guarda os dias da semana que o local fica aberto à visitação do público; endereco, para apresentar o endereço do lugar e preco, que é o valor que se paga na entrada do estabelecimento.
- Usuario: classe que guarda todas as informações referentes ao usuário. Os atributos identificados são: credits e creditsInternet, que guardam a quantidade de créditos para ligação e mensagem e para internet, respectivamente, que o usuário tem em conta; linguasFaladas, que mantém as línguas que o usuário conhece e conteudosInteresse, que guarda os conteúdos escolhidos pelo usuário. Os métodos levantados foram: pegarLinguasFaladas(), que retorna o atributo linguasFaladas, e pegarSaldo(), que retorna o atributo credits.
- Interprete: classe que contém as informações do intérprete. Seus atributos identificados são: idiomasFalados, que contém todos os idiomas que o intérprete domina; horasDisponivel e diasDisponivel, que juntos informam a disponibilidade do intérprete tendo o horário e os dias da semana, respectivamente, que ele pode atender.
- Questionario: classe que guarda as informações sobre o questionário de atendimento. Seus atributos são perguntas e respostas, que armazenam as perguntas cadastradas para o questionário e as respectivas respostas do usuário. Seu primeiro método é Questionario(protocolo : int), que é um método construtor da classe, que é chamado ao criar um objeto da classe e associa o protocolo do atendimento ao questionário. Outros métodos são pegarPerguntas(), que retorna o atributo perguntas, e adicionarRespostas(respostas : String[]), que salva as respostas do usuário no questionário.
- Duvida: classe abstrata que contém as informações em comum entre Ligacao e Mensagem. Seus atributos são: protocolo, o protocolo de atendimento; tempoUsado, tempo que o usuário já gastou dentro do limite e tempoLimite, período máximo que o usuário pode ficar tirando dúvidas. Esse tempo máximo é diferente para mensagens e para ligação, mas os dois tem tempo máximo.

- **Ligação:** classe que representa o modo de tirar dúvidas como ligação e herda de Duvida. O único método identificado é o `liberarLigacao()`, quando o Sistema permite a comunicação do usuário com o intérprete.
- **Mensagem:** classe que representa o modo de tirar dúvidas por mensagem. Seus atributos são: `conteudoDuvida`, que guarda a duvida do usuário; `respostaLinguaEstrangeira` e `respostaLinguaMaterna`, que armazenam as duas respostas do intérprete, na língua que o usuário entende e na que ele está com dúvida. Como a classe `Questionario`, `Mensagem` também tem um construtor que recebe como parâmetro o numero do protocolo e o conteúdo digitado pelo usuário. O outro método da classe é `guardarResposta(respostaEstrangeira : String, respostaMaterna : String)`, que salva no objeto as duas respostas do intérprete.

Já as classes de fronteira identificadas são todas as interfaces com os sistemas externos e com a tela: `InterfaceSistemaBancario`, `InterfaceSistemaDicionarioGirias`, `InterfaceSistemaPrevisaoTempo`, `InterfaceFusoHorario`, `InterfaceSistemaInformacoesPontosTuristicos`, `InterfaceUsuario`, `iOSInterfaceUsuario`, `AndroidInterfaceUsuario`, `InterfaceInterprete`, `iOSInterfaceInterprete`, `AndroidInterfaceInterprete`, `MensagemMapeadora`, `PacoteMapeadora`, `DicionarioMapeadora`, `QuestionarioMapeadora`, `MensagemNuvemMapeadora` e `PacoteNuvemMapeadora`. As interfaces `InterfaceMapeadora` e `InterfaceNuvemMapeadora` são fronteiras também e por isso serão descritas junto. Suas descrições seguem:

- **InterfaceSistemaBancario:** classe que se comunica com o Sistema Bancário para as transações com o cartão de crédito, caso o usuário queira comprar créditos através do aplicativo. O único método identificado foi `iniciarTransacao(valor : float)`, o que faz toda a comunicação com o sistema externo.
- **InterfaceSistemaDicionarioGirias:** classe que se comunica com o Sistema de Dicionário e Gírias. Seus métodos são `pegarPalavrasDicionario(lingua : String)`, para retornar as palavras com seus significados para o aplicativo, e `pegarGirias(cidade : String)`, que retorna as gírias locais de determinada cidade.
- **InterfaceSistemaPrevisaoTempo:** classe que se comunica com o Sistema de Previsão do Tempo, para pegar as informações atualizadas sobre o clima de determinada cidade. Seus métodos levantados são `atualizarClimaDiario(local : String, data : Date)`, método privado que pega o clima diário no sistema

externo, e atualizarClimaPeriodo(local : String, dataInicio : Date, dataFim : Date), um método público que chama o método privado para cada dia do período especificado.

- InterfaceFusoHorario: classe que se comunica com o Sistema de Fuso Horário, para calcular a diferença do fuso horário do local do pacote e o da cidade do usuário. Seus métodos são pegarFusoHorario(local : String), método privado que pega o fuso horário de um local específico, e pegarFusoHorario(localPacote : String, localNatal : String), que faz o cálculo da diferença.
- InterfaceSistemaInformacoesPontosTuristicos: classe que se comunica com o Sistema de Pontos Turísticos para pegar as informações dos pontos turísticos. Seus métodos são pegarHorario(cidade : String), pegarDiasAberto(cidade : String), pegarEndereco(cidade : String) e pegarPreco(cidade : String), todos métodos que pegam alguma informação sobre todos os pontos turísticos de determinada cidade.
- InterfaceUsuario: classe abstrata que guarda as assinaturas dos métodos comuns às classes que implementam a tela do usuário para diferentes sistemas operacionais. Seus métodos são: escolherTirarDuvidas(), que cria a tela inicial de tirar dúvidas; apresentarLinguasDisponiveis(linguasDisponiveis : String[]), que apresenta para o usuário quais das línguas conhecidas por ele tem intérprete disponível para tirar suas dúvidas; mostrarTelaMensagem(), que apresenta a tela para o usuário digitar sua dúvida; mostrarMensagemResposta(), que mostra para o usuário a resposta dupla do intérprete; perguntarMaisDuvidas(), que apresenta as opções de terminar atendimento ou continuar tirando dúvidas para o usuário escolher; mostrarQuestionario(perguntas : String[]), que mostra as perguntas que o usuário tem que responder ao final do atendimento, e apresentarInformacoesFinais(informacoes : String[]), que mostra na tela o saldo restante do usuário e o protocolo do atendimento. Vale ressaltar que todos esses métodos são abstratos, ou seja, só tem a assinatura dos mesmos nessa classe. Sua implementação fica a cargo das classes que herdarem InterfaceUsuario.
- iOSInterfaceUsuario: classe que herda de InterfaceUsuario e implementa seus métodos para o sistema operacional iOS.
- AndroidInterfaceUsuario: classe que herda de InterfaceUsuario e implementa seus métodos para o sistema operacional Android.

- **InterfaceInterprete**: classe abstrata que guarda as assinaturas dos métodos em comum às classes que implementam a tela do intérprete para diferentes sistemas operacionais. Seus métodos são: `avisaInterprete()`, que manda uma notificação para o interprete avisando que tem um usuário querendo tirar dúvidas; `cancelaAviso()`, que cancela a notificação enviada ao intérprete quando um outro aceitou o atendimento; `mostrarDuvida(conteudo : String)`, que mostra a dúvida do usuário e a tela para o intérprete responder, e `avisaTerminoAtendimento()`, que avisa ao intérprete que o atendimento foi encerrado. Como a classe `InterfaceUsuario`, todos os métodos são abstratos somente com a assinatura especificada.
- **iOSInterfaceInterprete**: classe que herda de `InterfaceInterprete` e implementa seus métodos para o sistema operacional iOS.
- **AndroidInterfaceInterprete**: classe que herda de `InterfaceInterprete` e implementa seus métodos para o sistema operacional Android.
- **MensagemMapeadora**: classe que se comunica com o Banco de Dados para persistir as mensagens que o usuário queira salvar. Seus métodos são descritos na interface `InterfaceMapeadora`, implementada na classe em questão.
- **PacoteMapeadora**: classe que se comunica com o Banco de Dados para persistir todos os pacotes criados. Seus métodos são descritos na interface `InterfaceMapeadora`, que a classe em questão implementa.
- **DicionarioMapeadora**: classe que se comunica com o Banco de Dados para persistir os dicionários criados. Seus métodos são descritos na interface `InterfaceMapeadora`, implementada pela classe em questão.
- **QuestionarioMapeadora**: classe que se comunica com o Banco de Dados para persistir os questionários respondidos. Seus métodos são descritos na interface `InterfaceMapeadora`, implementada pela classe em questão.
- **InterfaceMapeadora**: interface que persiste objetos no Banco de Dados. Seus métodos são `salvar(objeto : Object)`, que recebe um objeto da classe `Object`, na qual todas as classes são herdadas por padrão; e `pegar()`, que retorna um objeto da classe `Object`.
- **MensagemNuvemMapeadora**: classe que sabe persistir as mensagens que o usuário queira salvar na nuvem. Seus métodos são descritos na interface `InterfaceNuvemMapeadora`, implementada na classe em questão. Essa classe também se comunica com o Sistema de Armazenamento na Nuvem através da mesma interface `InterfaceNuvemMapeadora`.

- PacoteNuvemMapeadora: classe que sabe persistir os pacotes que o usuário comprar na nuvem. Seus métodos são descritos na interface InterfaceNuvemMapeadora, implementada na classe em questão. Essa classe também se comunica com o Sistema de Armazenamento na Nuvem através da mesma interface InterfaceNuvemMapeadora.
- InterfaceNuvemMapeadora: classe que se comunica com o Sistema de Armazenamento na Nuvem para persistir objetos salvos pelo usuário. Seus métodos são salvar(objeto : Object), que recebe um objeto da classe Object, na qual todas as classes são herdadas por padrão; e pegar(), que retorna um objeto da classe Object.

Por último, as classes de controle identificadas são: ControladorPacote, ControladorBD, ControladorNuvem, ControladorUsuario, ControladorTirarDuvidas, ControladorMensagem, ControladorInterprete e ControladorQuestionario. Todos os controladores são de fachada, exceto o ControladorTirarDuvidas que faz referência ao caso de uso Tirar Duvidas. Suas descrições vem a seguir:

- ControladorPacote: classe que faz a ligação entre as classes de fronteira dos sistemas externos que fazem referência às informações de Pacote, Cidade, PontoTuristico ou Dicionário. Essa classe também controla a classe Pacote. Seus métodos são iguais as classes de fronteira dos sistemas externos que se relacionam com o ControladorPacote, mais o método salvar(pacote : Pacote), que salva os pacotes completos no Banco de Dados e o salvarNuvem(pacote : Pacote), que salva os pacotes na nuvem.
- ControladorBD: classe que faz a ponte entre a InterfaceMapeadora e os objetos que precisam ser persistidos no Banco de Dados. Seus métodos são: obterInstancia(), que garante que só existirá uma única instância dessa classe; pegar(), que pega o objeto salvo no Banco de Dados; e salvar(objeto : Object), que salva o objeto em questão no Banco de Dados.
- ControladorUsuario: classe que faz a ponte entre a classe de fronteira da tela do usuário com o sistema e que pega e atualiza as informações da classe Usuario. Seus métodos são: iniciarTransacao(valor : float), igual a classe de fronteira do Sistema Bancário para comprar créditos; tirarDuvida(linguaDuvida : String), que retorna as línguas faladas pelo usuário que tem intérprete disponível; escolherModo(modos : String, linguas : String[]), que determina a escolha do usuário entre ligar para o intérprete ou mandar mensagem; terminarAtendimento(), quando o usuário não quer mais tirar dúvidas; enviarRespostasQuestionario(respostas : String[]), que envia as respostas do

usuário para serem salvas no Questionario; pegarMensagem(), que pega a mensagem de dúvida do usuário; perguntarMaisDuvidas(), que chama o método perguntarMaisDuvidas() de InterfaceUsuario; verificarSaldo(saldo : int), método privado que verifica se o usuário tem créditos suficientes para continuar tirando dúvidas; pegarLinguasFaladas(), que pega todas as linguas faladas pelo usuário; verificaSaldo(), método público para pegar o saldo do usuário e chamar o método privado de mesmo nome; mostrarQuestionario(perguntas : String[]) que chama o método de mesma assinatura de InterfaceUsuario; mostrarInformacoesFinais(protocolo : String), que pega o saldo e chama o método de mesmo nome de InterfaceUsuario; mostrarTelaMensagem(), que chama o método de mesmo nome de InterfaceUsuario; e repassarRespostaMensagem(conteudoMaterno : String, conteudoEstrangeiro : String), que chama o nome de mesma assinatura de InterfaceUsuario.

- ControladorTirarDuvidas: classe de controle principal do caso de uso escolhido que se comunica com os outros controladores para a continuidade do caso de uso. Seus métodos são: tirarDuvidas(linguaDuvida : String), que pega as línguas que o usuário sabe e procura os intérpretes que saibam a língua em dúvida e pelo menos uma língua do usuário e retorna quais das línguas do usuário que tem algum intérprete; escolherModo(modos : String, linguas : String[]), que determina qual o modo que o usuário tirará dúvidas; terminarAtendimento(), que avisa ao intérprete que terminou o atendimento e pega as perguntas do Questionario; enviarRespostasQuestionario(respostas : String[]), que manda as respostas do usuário para serem salvas no Questionario; pegarMensagem(), que pega a mensagem de dúvida do usuário; e repassarRespostaMensagem(conteudoMaterno : String, conteudoEstrangeiro : String), que manda a dupla resposta do intérprete para o usuário.
- ControladorMensagem: classe que controla a troca de mensagem do sistema. Seus métodos são: escolherModoMensagem(linguas : String[], interpretes : Interprete[]), que notifica todos os intérpretes e espera um aceitar o atendimento; gerarProtocolo(), que gera o protocolo de atendimento; terminarAtendimento(), para avisar aos intérprete que o atendimento acabou; repassarResposta(conteudoMaterno : String, conteudoEstrangeiro : String), que repassa a resposta do intérprete até chegar ao usuário; e salvarNuvem(mensagem : Mensagem), que salva na nuvem as mensagens.
- ControladorInterprete: classe que controla a classe Interprete e se comunica com InterfaceInterprete. Seus métodos são: notificarInterpretes(interpretes : Interprete[]), que avisa a todos os intérpretes que tem um usuário com dúvidas,

espera um responder e cancela o aviso para os outros;  
enviarMensagem(conteudo : String), que envia a mensagem para o intérprete;  
enviarResposta(conteudoEstrangeiro : String, conteudoMaterno : String), que repassa a resposta do intérprete; avisarTerminoAtendimento(), que chama o método de mesmo nome em InterfaceInterprete;  
pegarInterpretesDisponiveis(linguasFaladas : String[], linguaDuvida : String), que retorna os intérpretes disponíveis naquele horário que saiba falar a língua que o usuário tem dúvida e pelo menos uma das línguas que ele saiba falar.

- ControladorQuestionario: classe que controla a classe Questionario. Seus métodos são: terminarAtendimento(protocolo : int), que cria uma instância da classe Questionario e retorna as perguntas da mesma; guardarRespostas(respostas : String[]), que guarda as respostas do usuário no Questionario; e salvar(questionário : Questionario), que salva no Banco de Dados o Questionario.



## Capítulo VII - Diagrama de Sequência de 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.

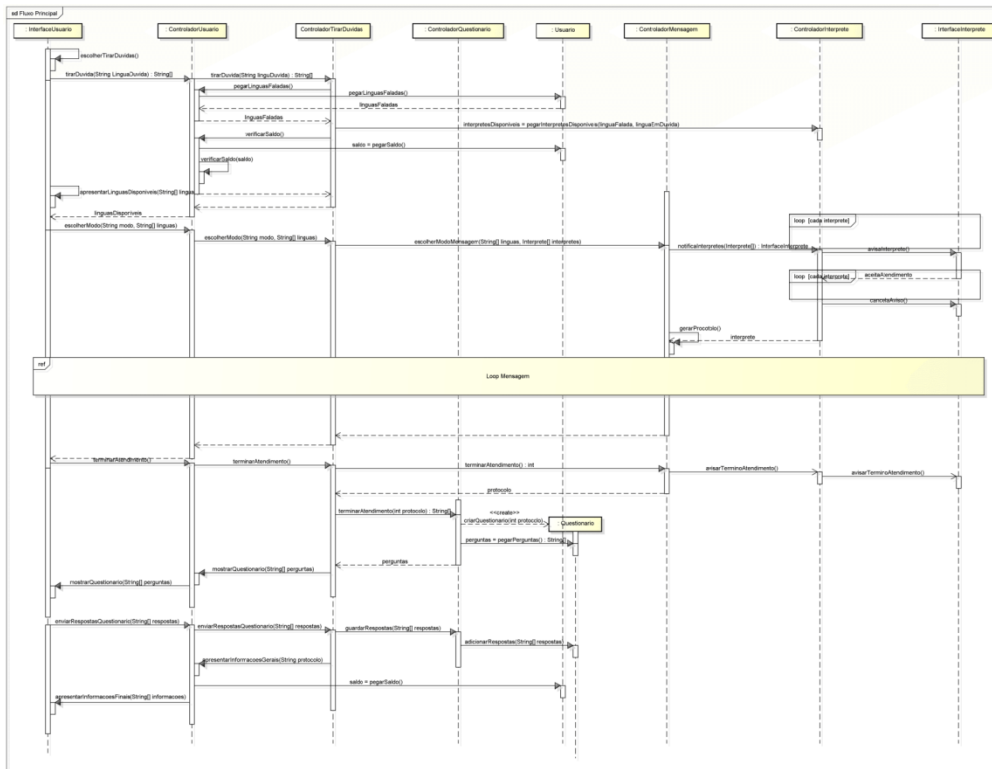
Neste diagrama, são criados diferentes objetos que podem ser nomeados, anônimos ou uma coleção. Objeto nomeado é aquele ao qual foi dado um nome. Normalmente um objeto nomeado é utilizado quando precisa ser referenciado em mais de um lugar. Objeto anônimo é aquele ao qual não foi dado nenhum nome. E 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 (BEZERRA, Eduardo)<sup>3</sup>.

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. Outros tipos também são representados, sendo eles: de criação (setas com linhas tracejadas), cuja linha de vida inicia na posição em que é criado, além de apresentar o estereótipo <<create>> na seta; de destruição (setas fechadas e preenchidas), que apresenta um símbolo X no final da linha de vida dizendo que o objeto está sendo destruído, além de apresentar o estereótipo <<destroy>> na seta; perda (seta fechada e preenchida seguida de uma bola preenchida), quando a mensagem não chegou ao seu destino e encontrada (com uma bola preenchida seguida de uma seta fechada e preenchida), quando a origem está fora do escopo da descrição.

### VII.1 - Diagrama de Sequência Fluxo Principal

---

<sup>3</sup> Princípios de Análise e Projeto de Sistemas com UML. Capítulo 7



**Figura 13 - Diagrama de Sequência Fluxo Principal**

### VII.1.1 - Descrição do Diagrama de Sequência Fluxo Principal

Este diagrama representa o Fluxo Principal do caso de uso estendido Tirar Dúvidas. Para começar, a InterfaceUsuario disponibiliza para o usuário a opção de tirar dúvidas. Quando o usuário escolher essa opção, uma mensagem é enviada para o ControladorUsuario com o parâmetro da língua que o usuário escolheu para tirar dúvidas e, em seguida, para o ControladorTirarDuvidas. Este último controlador pega do ControladorUsuario e, em seguida, do Usuario as línguas que o mesmo fala. Assim, o ControladorTirarDuvidas pega do ControladorInterprete os intérpretes que estão disponíveis, passando como parâmetros a língua que o usuário escolheu para tirar dúvidas e as línguas que o mesmo fala. Durante esse processo, o ControladorUsuario verifica a quantidade de saldo do usuário e após a verificação, a InterfaceUsuario apresenta as línguas disponíveis, conforme as línguas que o usuário domina.

Seguindo o processo, a InterfaceUsuario apresenta a opção escolher modo, onde o usuário terá duas maneiras de comunicação, uma via mensagem e outra via telefonema. Porém descreveremos o modo mensagem nesse fluxo principal. Sendo assim, o usuário informa o modo de comunicação mensagem e as línguas que iniciarão a conversa. O ControladorTirarDuvida envia uma mensagem com as línguas e os intérpretes disponíveis para o ControladorMensagem. Este notifica o ControladorInterpretes sobre um usuário com dúvidas, sendo assim um aviso é

enviado a todos os intérpretes. Quando um intérprete aceitar a comunicação, uma mensagem de cancelamento é enviada para os outros intérpretes. Portanto, o ContraladorInterprete avisa o ControladorMensagem que um intérprete aceitou tirar dúvidas do usuário. Com isso, um protocolo é gerado e um LoopMensagem é iniciado. Este LoopMensagem será descrito a seguir, separadamente.

Com o fim do atendimento, um aviso é enviado ao InterfaceIntérprete. E termina a geração de protocolo, na qual o ControladorQuestionario recebe uma mensagem sobre o fim do atendimento informando o protocolo como parâmetro. É criada uma instância Questionario e depois da criação, ele pega as perguntas e envia ao ControladorTirarDuvidas, que mostra esses dados na InterfaceUsuario. O usuário responde e as respostas são enviadas até serem adicionadas no Questionario. No fim do processo, o ControladorTirarDuvidas envia uma mensagem para apresentar as informações gerais do atendimento para o usuário. O ControladorUsuario recebe essa mensagem e, em seguida, pega o saldo restante do usuário. Posteriormente, essas informações são mostradas para o usuário na InterfaceUsuario.

## VII.2 - Diagrama de Sequência Loop Mensagem

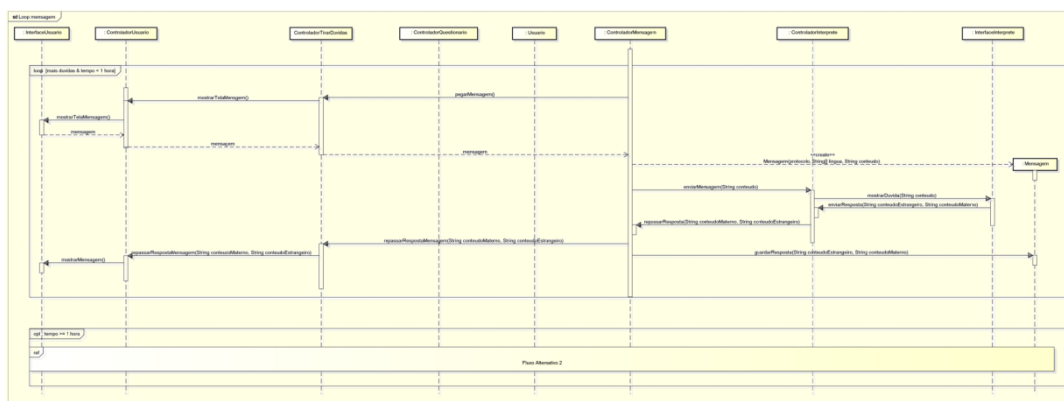


Figura 14 - Diagrama de Sequência Loop Mensagem

### VII.2.1 - Descrição do Diagrama de Sequência Loop Mensagem

Este diagrama é uma referência ao diagrama Fluxo Principal 1, onde é feita toda a comunicação entre o usuário e o intérprete. Para iniciar o loop e a troca de mensagens, o ControladorMensagem notifica ao ControladorTirarDuvidas que quer iniciar uma conversa via mensagem e o ControladorTirarDuvidas mostra a tela mensagem na InterfaceUsuario. Este digita suas dúvidas, que são enviadas para o ControladorTirarDuvidas que encaminha para o ControladorMensagem. Este cria uma nova instância Mensagem com o protocolo, línguas e a mensagem, com todos os possíveis parâmetros para conter as novas mensagens trocadas. Logo o

ControladorMensagem envia para o ControladorInterprete o conteúdo e estes dados são mostrados na InterfaceInterprete. O intérprete responde as dúvidas na língua materna do usuário e na língua estrangeira, e o ControladorInterprete repassa as respostas para o ControladorMensagem que, em seguida, são repassadas para o ControladorTirarDuvidas e por consequência, para o ControladorUsuario. Estes repasses visam sempre diminuir o acoplamento, repassando até o ControladorUsuario mostrar as mensagens na tela do usuário. Lembrando que o ControladorMensagem guarda as respostas em Mensagem.

Esse ciclo fica ativado até o usuário não querer tirar mais dúvidas ou até uma hora de duração, que é o tempo máximo de comunicação por mensagens. Logo após o término de troca de mensagens, o ControladorMensagem termina de executar a comunicação entre usuário e intérprete. Caso o usuário necessite trocar mais mensagens, mesmo chegando ao tempo limite de uma hora, um fluxo alternativo é iniciado para uma nova troca de informações. Este fluxo será descrito a seguir, em outro tópico.

### VII.3 - Diagrama de Sequência Fluxo Alternativo 2

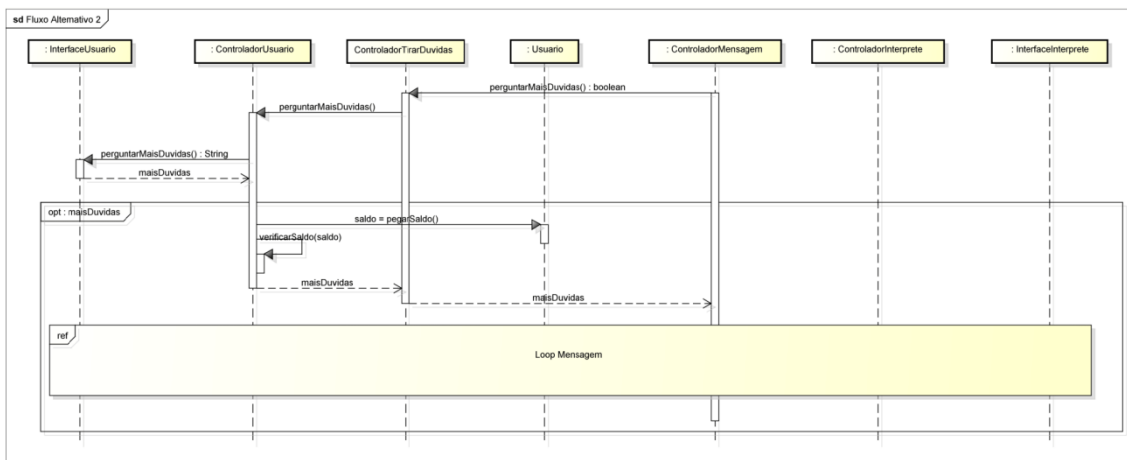


Figura 15 - Diagrama de Sequência Fluxo Alternativo 2

#### VII.3.1 - Descrição do Diagrama de Sequência Fluxo Alternativo 2

O fluxo alternativo 2 é iniciado com o ControladorMensagem perguntando para o ControladorTirarDuvidas se o usuário tem mais dúvidas e este envia a pergunta para o ControladorUsuario, que envia para a InterfaceUsuario. Se o usuário responder que sim, um novo fragmento encapsulado do tipo alt é criado e o saldo do usuário é verificado. Se o saldo for suficiente o Loop Mensagem é iniciado. Este já foi descrito anteriormente.

## Capítulo VIII - Diagrama de Atividades

No diagrama de atividades são apresentadas as atividades de um fluxo de trabalho que corresponde a um comportamento do sistema. O diagrama é orientado a fluxo de controle que é simbolizado por setas direcionadas e elementos que descrevem as atividades. Uma atividade pode ser realizada imediatamente ou levar um tempo para ser finalizada.

O diagrama tem estados inicial e final. O estado inicial é representado por um círculo preenchido que dá início a uma ação e é único e obrigatório. O estado final é representado por dois círculos, com o círculo central preenchido, que indica o fim do fluxo de controle e pode ter mais de um no diagrama. Caso o diagrama não apresente o estado final, significa que foi modelado um processo cíclico. Cada atividade é representada por um retângulo com bordas arredondadas e quando uma atividade é concluída, a próxima se inicia automaticamente.

Uma dessas transições pode chegar a um ponto de ramificação que mostra caminhos distintos para a ocorrência de uma ação ou atividade. Em cada caminho há uma condição de guarda que avalia uma expressão a partir do ponto ramificação. O ponto de ramificação é representado por um losango. De todas as possibilidades, apenas uma condição de guarda deve ser verdadeira. Pode haver uma representação de condição de guarda [else] caso as demais condições forem avaliadas para falso. Diversos fluxos podem ser unidos a um ponto de união, um losango onde chegam todos esses fluxos, mas sai somente um.

Um diagrama de atividade pode conter fluxos de controle paralelos. Isso significa que pode haver dois ou mais fluxos de controle sendo executados simultaneamente em um diagrama de atividades (BEZERRA, Eduardo)<sup>4</sup>. São utilizadas barra de bifurcação (fork) e barra de junção (join). A barra de bifurcação é usada quando a mesma recebe um fluxo de controle e em seguida, cria dois ou mais fluxos paralelos, sendo executados independentemente. Já a barra de junção é usada quando a mesma recebe dois ou mais fluxos de controle e os une em um único fluxo. Este caso ocorre quando todos os fluxos tiverem alcançado os pontos de junção.

As atividades deste diagrama podem ser realizadas por diferentes agentes. Nesse caso, pode ser representado no diagrama raias de natação, uma para cada agente. Essas raias dividem o diagrama em compartimentos e as atividades que serão realizadas por determinado agente ficam na raia correspondente.

Nesse diagrama, também pode ser utilizado objetos durante o fluxo,

---

<sup>4</sup> Princípios de Análise e Projeto de Sistemas com UML. Capítulo 11.

representados por um retângulo, a fim de mostrar o estado deste objeto diante do fluxo de uma ação ou atividade para a outra. Eles podem ser produzidos ou alterados por ações ou atividades. Além disso, também pode ser criada uma região de atividade interrompível que estabelece uma ação que pode ser interrompida a qualquer momento.

### **VIII.1 - Diagrama de Atividades**

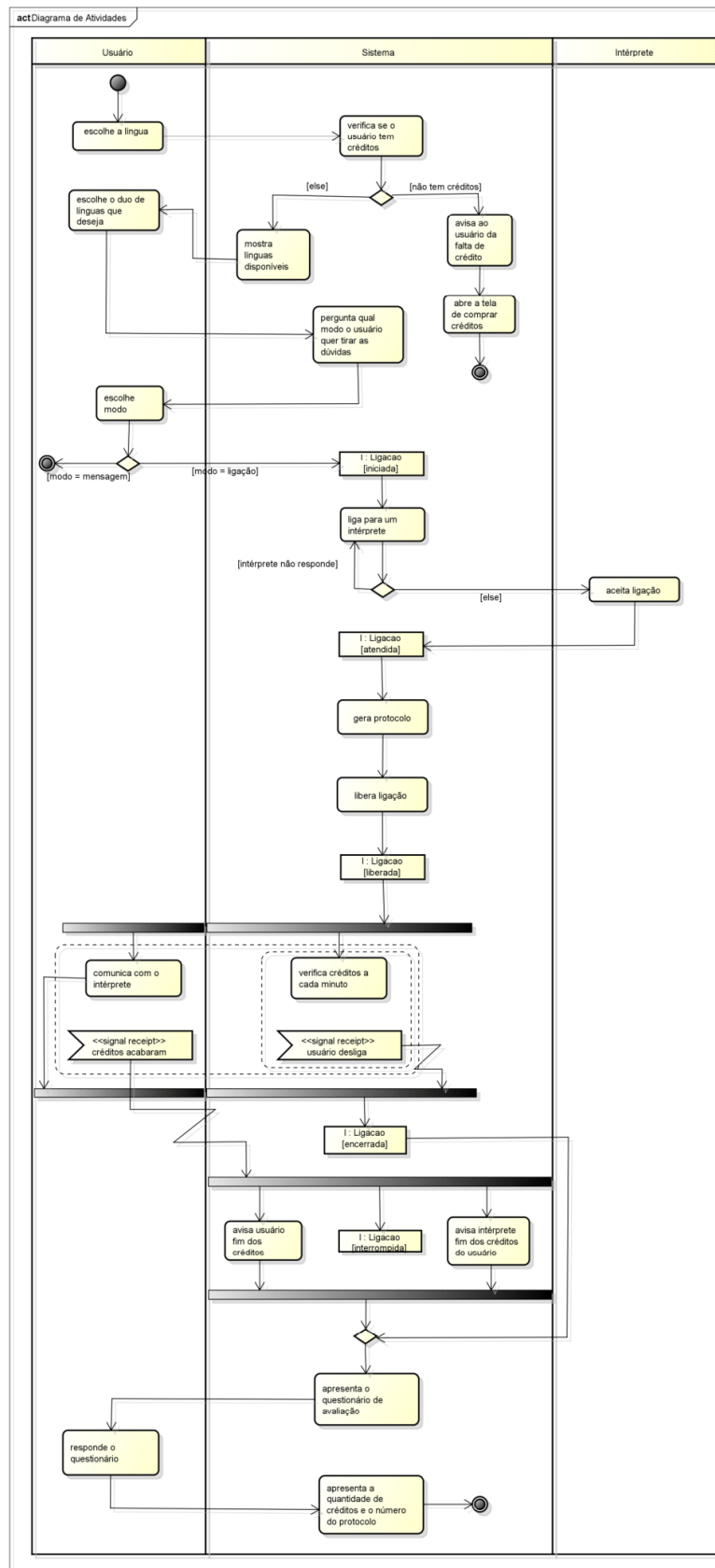


Figura 16 - Diagrama de Atividades

### VIII.1.1 - Descrição do Diagrama de Atividades

Utilizamos neste diagrama de atividades somente o fluxo alternativo 1 (Ligação), fluxo de exceção 1 e 2. E possui as raias de natação do Usuário, Sistema e Intérprete. Para começar, o usuário escolhe a língua que deseja tirar dúvidas. O sistema verifica se o usuário tem créditos para o mesmo. Caso não tenha, o sistema avisa ao usuário da falta de créditos e vai aparecer na tela do usuário a opção de comprar créditos e o fluxo termina. Caso o usuário tenha créditos, o sistema mostra as línguas que estão disponíveis para se tirar as dúvidas.

Logo em sequência, o usuário escolhe o duo de línguas desejado para prosseguir sua ação. O sistema então pergunta qual o modo o usuário deseja tirar suas dúvidas, ligando ou mandando mensagem. O usuário escolhe o modo. Se o modo for mandar mensagem, o fluxo terminará. Se for ligação, o objeto do tipo Ligacao vai para o estado iniciada no sistema, permitindo com que o sistema ligue para um intérprete. Se um intérprete não responder, o sistema continua ligando para um até que um deles aceite a ligação. Quando isso acontece, o objeto do tipo Ligacao vai para o estado atendida e logo após, o sistema gera o protocolo.

Com isso, o sistema libera então a ligação e o objeto do tipo Ligacao vai para o estado liberado. Uma ramificação é criada e ao mesmo tempo que ocorre a comunicação do usuário com o intérprete, a verificação dos créditos a cada minuto é feita pelo sistema.

Se o usuário desligar terminando a atividade, a verificação dos créditos é interrompida, fechando a ramificação, e gerando um objeto do tipo Ligacao que vai para o estado encerrada. Caso os créditos terminem, as duas atividades são interrompidas, abrindo outro ponto de bifurcação que ao mesmo tempo o objeto do tipo Ligacao vai para o estado interrompida, o sistema avisa o usuário que seus créditos chegaram ao fim e avisa também o intérprete do fim dos créditos do usuário. A ramificação é fechada e ligada a uma um ponto de união, permitindo com que o sistema apresente o questionário de avaliação e que o mesmo seja respondido pelo usuário. Em seguida, o sistema apresenta quantidade de créditos e o número de protocolo para o usuário e é encerrado o diagrama de atividades.



## Capítulo IX - Diagrama de Estados

O diagrama de estados descreve o ciclo de vida de objetos de uma única classe do sistema. Este ciclo é composto por transições de estados, e o objeto, que representa o mundo real, se encontra em um estado particular. O estado de um objeto caracteriza uma situação da vida em que ele realiza atividades ou ações, e a mudança desse estado para outro, mais conhecida como transição entre estados, acontece quando ocorre algum evento interno ou externo ao sistema. Uma ação é representada na linha de transição e por uma barra inclinada, que é declarada por uma expressão que associe a transição de estados. A atividade indica alguma tarefa que é executada pelo objeto. Enquanto que a ação está associada a uma transição, a atividade está associada a um estado.

São representados elementos que definem o diagrama de estados. Nele, temos os estados inicial, final, ordinário, aninhado e concorrente. O estado inicial, por onde o diagrama começa a ser lido, é representado por um círculo preenchido que indica o estado de um objeto quando ele é criado. O estado final é representado por dois círculos, com o círculo central preenchido, que indica o estado final do ciclo de vida de um objeto. Como no diagrama de atividades, o estado inicial é obrigatório e o final opcional. O estado ordinário é representado por um retângulo com bordas arredondadas que indica o estado em que o objeto se encontra. Os estados aninhados tem vários estados dentro, mas pode ser visto como um único estado. Nesse caso, pode haver mais de um estado inicial, um do diagrama e um dentro de cada estado aninhado.

A transição entre estados é representada por uma linha conectando estados, e ela também pode ser transição reflexiva, quando uma transição leva ao mesmo estado. Na transição, pode definir uma expressão que indica um evento ou uma ação para aquela transição. O evento acontece em determinado tempo e pode modificar o estado de um objeto. Os eventos podem ser de chamada, quando uma mensagem é solicitada e espera pela resposta; temporal, quando ocorre uma passagem de intervalo de tempo; de sinal, quando é enviado um sinal para outro objeto para dar continuidade a sua execução; e de mudança, quando uma condição se torna verdadeira. Numa transição pode ter uma condição de guarda que é representada entre colchetes. Mais conhecida como guarda, ela define uma condição que determina a execução do evento, sendo somente quando a mesma é verdadeira; caso contrário, o evento é ignorado. Uma escolha, representada por um losango, ocorre quando uma transição se divide em múltiplas transições, onde cada transição é associada a uma condição de guarda. Já uma junção, representada por um círculo preenchido, acontece quando

duas ou mais transições apresentam uma ou mais saídas de transição com uma condição de guarda. Quando uma transição se divide em duas ou mais transições de saída, é chamada de fork. Caso várias transições se juntam em uma única transição de saída, é chamada de join. Ambos os casos são divididos por uma barra vertical.

Além da transição de estado, também existe a transição interna, que acontece quando uma ação ou atividade é realizada sem alterar o estado do objeto. Nesse caso, cláusulas reservadas são utilizadas para indicar as transições que ocorrem dentro de um estado, sem que o mesmo seja alterado no final da transição. A cláusula *entry* mostra a ação executada na entrada do estado, o *do* é uma atividade que ocorre enquanto o objeto está no estado e o *exit* mostra a ação executada na saída do estado.

### IX.1 - Diagrama de Estado

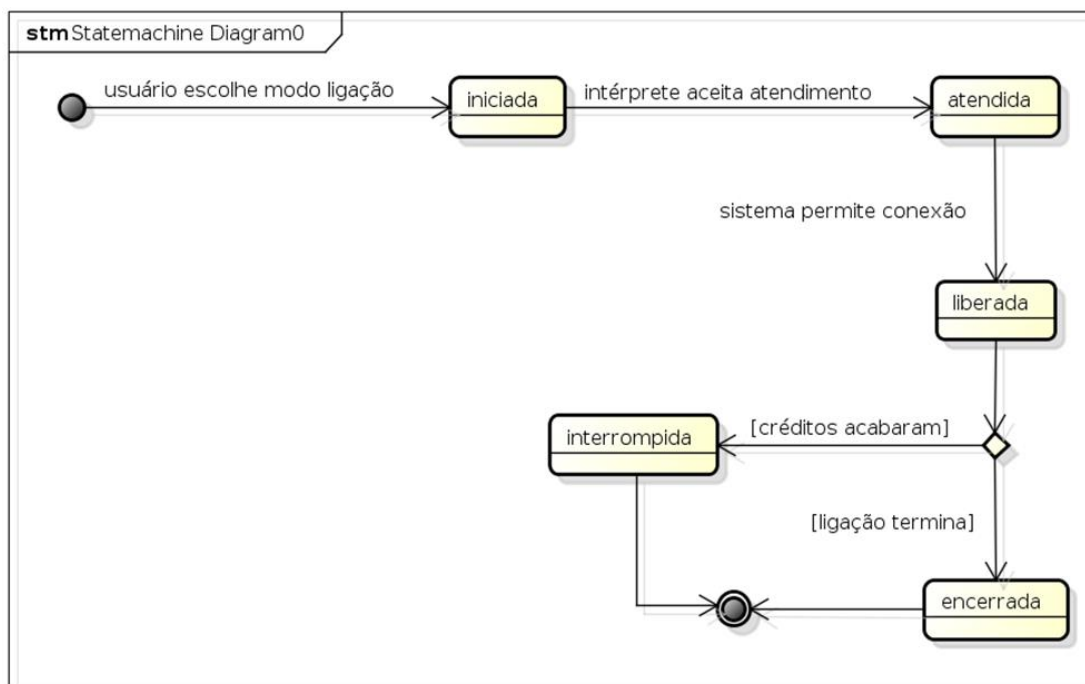


Figura 17 - Diagrama de Estado

#### IX.1.1 - Descrição do Diagrama de Estado

O diagrama de transição de estados descreverá o ciclo de vida da classe *Ligacao*. Para iniciar o usuário escolhe o modo ligação. Sendo assim, o estado passa a ser começado. A transição para o estado *atendida* é feita quando um intérprete aceita o atendimento. Com isso, a ação *sistema permite conexão* é efetuada e o estado passa a ser *liberada*. Com o estado *liberada*, as transições seguintes são divididas em múltiplas escolhas: *interrompida*, que tem a condição de guarda *créditos*

acabaram e encerrada, que tem a condição de guarda ligação termina. O estado final do diagrama de estados ocorre quando os estados forem interrompida ou encerrada.

## **Capítulo X - Conclusão**

Começar o trabalho foi o primeiro obstáculo que encontramos. Como pensar numa extensão complexa de um aplicativo já existente? Porém, com esforço chegamos a um acordo, escolhemos desenvolver o MyUniverse, que será uma funcionalidade de guia turístico.

Durante o desenvolvimento nos deparamos com a falta de tempo, conciliação com outras disciplinas, divisão das tarefas e o mais importante a dificuldade de elaborar os diagramas.

Todavia aprendemos bastante com a prática em equipe, que é uma qualificação importante no mercado de trabalho. Também são poucas disciplinas que deixam o lado teórico e dão espaço para colocar em prática o conteúdo apresentado em sala. Com isso, chegamos a conclusão que a modelagem dos diagramas, aplicando o padrão UML, é fundamental para a elaboração de bons projetos.

Esperamos levar toda essa experiência adquirida nessa disciplina para o mercado de trabalho. E assim solidificar o que aprendemos e ser um diferencial.

### **Referências Bibliográficas**

BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 3ª Tiragem. Editora Campus.

LARMAN, Craig. Utilizando UML e Padrões. 3ª Edição. 2005.

## Apêndice A - Lista de Categorias

Tabela 2 – Listas de Categorias

<b>Categoria ou Classe Conceitual</b>	<b>Exemplos</b>
Transações de negócio (envolvem dinheiro)	ComprarCreditos
Transações de itens	ItemDeCrédito
Produtos ou serviços relacionados a transação ou item	Créditos
Onde a transação é registrada?	Sistema
Papeis de pessoas ou organizações	Usuário, Intérprete
Local de transação ou serviço	Lojas, SiteDePagamentos
Eventos notáveis (ex. com hora e local)	ComprarCréditos
Descrição de coisas	DescriçãoDaDúvida, DescriçãoDoPacote, PerfilDoUsuário
Catálogos	CatálogoDePacotes
Contêiner de coisas (físico ou informação)	Pacote, Nuvem
Coisas em um contêiner	Dicionário, PontosTurísticos, Clima, GíriasDoLocal, TelefonesEmergências, FusosHorários, Mensagem
Outros sistemas	Clima, Dicionário, PontosTurísticos, FusosHorários
Registros de finanças, trabalho, contrato, assuntos legais	Recibo, Questionário
Instrumentos financeiros	LinhaDeCrédito
Horários, manuais, documentos consultados para realizar trabalho	ListaDePontosTurísticos, Dicionário, TelefonesEmergenciais, Clima

## Apêndice B - Análise Gramatical

Tabela 3 – Análise Gramatical

	Nome	Significado	É uma classe?	Justificativa
1	Aplicativo	Aplicativo MyUniverse, que faz a ponte entre o interpretador e o usuário.	Não	Consiste no próprio sistema.
2	Cadastro	Conjunto de informações pessoais do usuário, levando em consideração as línguas de preferência do mesmo.	Não	Está contido dentro da classe Usuário.
3	Caso de uso 'comprar créditos'	Modo de obter créditos para ligar ou mandar mensagens para o interpretador.	Não	É uma ação/método na classe Usuário.
4	Compra do pacote/pacote comprado	Método de escolher o pacote conforme o país desejado pelo usuário.	Não	Seria um método da classe Pacote.
5	Comunicação	Maneira de diálogo entre duas pessoas ou entre uma pessoa e um sistema.	Não	Simplesmente um relacionamento entre o Usuário e o Intérprete.
6	Configuração	Consiste na opção de customizar o sistema conforme o interesse do usuário.	Não	Não provê nenhum serviço no sistema.
7	Conta/Conta de acesso	Meio de identificação do usuário pelo sistema após o cadastro.	Não	Encontra-se dentro da classe Usuário.
8	Conteúdo de interesse/preferência	Informações de interesse do usuário sobre as características da região que ele deseja visitar.	Não	Seria um atributo da classe Usuário.
9	Crédito de internet/crédito para acessar a nuvem	Custo de acesso à internet e a nuvem, que diminuem conforme a navegação.	Não	Simplesmente um atributo de Usuário.

<b>10</b>	Créditos restante/créditos em conta	Após ligações, mensagens e acesso a nuvem, sobrarão créditos, que ficarão na conta do usuário e poderão ser usados quando ele quiser.	Não	Atributo de Usuário.
<b>11</b>	Créditos suficientes	Saldo em dinheiro virtual para fazer ligações, mandar mensagens, comprar pacote e acessar a nuvem.	Não	Atributo de Pacote (comprar pacotes) e o restante são constantes.
<b>12</b>	Dados pessoais	Informações sobre o usuário que utilizará o sistema.	Não	Seriam atributos da classe Usuário.
<b>13</b>	Dicionário	Coleção organizada em ordem alfabética de palavras conforme a língua escolhida.	Sim	Gerencia dados complexos.
<b>14</b>	Diferença de fuso horário	Conta realizada subtraindo a hora local de onde o usuário mora da hora local da região do pacote.	Não	Atributo de Cidade.
<b>15</b>	Duo de línguas	As duas línguas que o usuário quer tirar dúvidas, a falada e a estrangeira.	Não	Atributo de Dúvida.
<b>16</b>	Dúvidas	O que o usuário não entende.	Sim	O principal conjunto de dados gerenciado pelo sistema.
<b>17</b>	Espaço da memória do celular	Onde os aplicativos e dados são armazenados no celular.	Não	Não está dentro do limite do sistema.
<b>18</b>	Execução do caso de uso 'tirar dúvidas'	Passos para o usuário sanar suas dúvidas.	Não	Não tem uma estrutura identificável, pois é um processo do sistema.



<b>19</b>	Forma como o usuário quer tirar as dúvidas	Modo que o usuário pode tirar suas dúvidas, através de ligação ou mensagem.	Não	Métodos de Dúvida.
<b>20</b>	Frases básicas para interação	Frases simples como "Oi, tudo bem?", "meu nome é...", entre outras.	Não	Atributo de Pacote.
<b>21</b>	Gírias da região	Palavras ou expressões usadas pelos nativos.	Não	Atributo de Cidade.
<b>22</b>	Hora	Tempo limite para troca de mensagens.	Não	Tempo limite para mensagem.
<b>23</b>	Idiomas falados/línguas faladas	Idiomas que o usuário domina.	Não	Simplesmente atributo do Usuário e do Intérprete.
<b>24</b>	Informação	Dado sobre determinada região.	Não	Não provê nenhum serviço.
<b>25</b>	Intérprete	Pessoa que fala duas línguas e tem disponibilidade para tirar dúvidas.	Sim	Gerencia todas as informações do Intérprete.
<b>26</b>	Intérprete disponível	Intérprete que esteja disponível no momento para falar com o usuário.	Não	Seria um estado de Intérprete.
<b>27</b>	Ligação	Recurso que permite a comunicação entre o sistema e o intérprete e entre o intérprete e o usuário.	Não	Um tipo de comunicação do sistema.
<b>28</b>	Língua	Instrumento de comunicação composto em diversos idiomas que auxilia na compreensão de enunciados entre duas ou mais pessoas.	Não	Atributo de Usuário e Intérprete.
<b>29</b>	Mensagem	Mensagem de texto que o usuário utiliza para enviar dúvidas para os intérpretes cadastrados no	Não	Um tipo de comunicação do sistema.

		sistema.		
<b>30</b>	Mensagem recebida do intérprete	Mensagem de texto recebida do intérprete que esclarece as dúvidas declaradas na mensagem que foi enviada pelo usuário.	Não	Um tipo de comunicação do sistema.
<b>31</b>	Momento	Refere-se ao determinado tempo, entre o intervalo de uma hora, em que o usuário finaliza a comunicação com o intérprete.	Não	Não está no limite do sistema.
<b>32</b>	Minuto de ligação	Intervalo de tempo em que o sistema verifica se o usuário ainda possui créditos para continuar a ligação com o intérprete.	Não	Tempo limite de ligação.
<b>33</b>	Nuvem	Local onde o usuário, com o acesso à Internet no celular, salva as mensagens recebidas do intérprete, além de salvar os pacotes escolhidos por ele automaticamente. Também permite visualizar estes conteúdos.	Não	Fora do contexto do sistema.
<b>34</b>	Opção escolher pacote/Opção visualizar pacotes/Opção tirar dúvidas	Itens presentes no aplicativo que permite o usuário escolher o pacote que deseja comprar, visualizar os pacotes comprados e tirar suas dúvidas por meio de ligação ou mensagem com o intérprete, respectivamente.	Não	Somente um menu visual para o usuário.
<b>35</b>	Opção salvar	Item presente no aplicativo que permite o usuário salvar na nuvem as mensagens recebidas do	Não	Seria o método salvar na classe Mensagem.

		intérprete.		
<b>36</b>	Pacote/quantidade de informações/dados	Apresenta conjunto de dados do país/região/cidade de destino, como telefones emergências, pontos turísticos, gírias locais, entre outros, comprados pelo usuário. Cada pacote pode conter uma ou mais dessas opções de acordo com a sua preferência.	Sim	Gerencia informações de uma determinada região.
<b>37</b>	Passeio	Atividade feita pelo usuário afim de visitar pontos turísticos do país estrangeiro, por exemplo.	Não	Não está no limite do sistema.
<b>38</b>	Perfil traçado	Consiste nas características que definem o usuário presentes no cadastro.	Não	Mesmo atributo de conteúdos de interesse em Usuário.
<b>39</b>	Primeiro acesso	Primeira vez que o usuário entra na opção 'Escolher Pacote'.	Não	Não manipula nenhum dado no sistema.
<b>40</b>	Primeiro pacote escolhido	Somente o primeiro pacote escolhido na hora do cadastro é gratuito.	Não	Não provê nenhum serviço no sistema.
<b>41</b>	Pontos turísticos	Informações que o sistema carrega da nuvem com o intuito de sugerir lugares para o usuário conhecer.	Sim	Gerencia as informações de cada ponto turístico.
<b>42</b>	Protocolo de atendimento/número do protocolo de atendimento	Identificação da troca de mensagem ou da ligação.	Não	Atributo de Dúvida.
<b>43</b>	Quantidade de créditos	Verificação que o sistema faz a cada minuto de ligação para permitir a comunicação entre o usuário e o intérprete.	Não	Atributo de Usuário.

44	Questionário de avaliação do atendimento/avaliação	Ao final do caso de uso 'tirar dúvidas', o sistema apresenta para o usuário um questionário a fim de obter um feedback do cliente.	Sim	Gerencia a avaliação do atendimento dado pelo intérprete.
45	Rede do celular	Configuração que permite com que o usuário utilize sua rede de celular (móvel ou wi-fi) para acessar seus pacotes.	Não	Atributo de Usuário.
46	Região do pacote/cidade	Local do pacote que o usuário escolheu na hora do cadastro.	Sim	Gerencia as informações de cada região.
47	Semana	Período em que o sistema envia sugestões para o usuário conforme o perfil traçado no momento das compras dos pacotes.	Não	Somente uma variação de tempo.
48	Sistema	É o aplicativo como um todo.	Não	Consiste no próprio aplicativo.
49	Sistema de previsão do tempo	Sistema que possui as informações atualizadas a respeito do clima diário e semanal das regiões dos pacotes do usuário.	Não	Não faz parte do contexto do sistema.
50	Sugestão	Dicas que o sistema envia toda semana para o usuário conforme o perfil traçado no momento das compras dos pacotes.	Não	Um meio de comunicação.
51	Tempo limite	Tempo limite de 1 hora de ligação que o usuário pode ter para tirar dúvidas com o intérprete.	Não	Atributo de Dúvida.
52	Término	Fim do tempo limite de 1 hora de ligação entre o usuário e o intérprete.	Não	Um estado de finalização.
53	Todo dia	Período em que o sistema	Não	Um estado de

		atualiza o clima das regiões dos pacotes do usuário.		atualização.
<b>54</b>	Usuário	Pessoa que utilizará o sistema.	Sim	Gerencia as informações do próprio usuário.
<b>55</b>	Violação	Uma forma de desrespeito a uma regra do sistema.	Não	Um estado de não conformidade com uma regra.