

Interactive Voice Response para Call Center de Táxis

Eduardo Filipe Leite Cunha Lima

Mestrado Integrado Engenharia de Redes e Sistemas Informáticos

Departamento de Ciências de Computadores

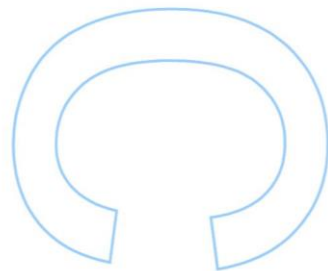
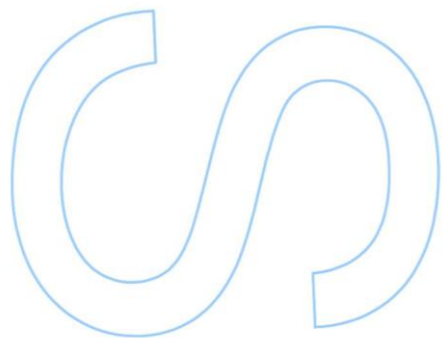
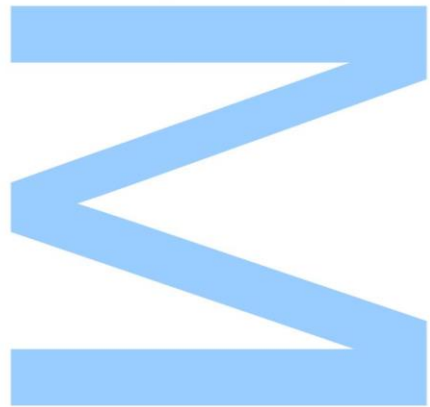
2014

Orientador

Dr. José Alves

Coorientador

Professor Doutor Michel Ferreira, DCC-FCUP



Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, ____/____/____

3

S

Q

AGRADECIMENTOS

À minha família, e à minha namorada, um enorme agradecimento pelo apoio incondicional a todos os níveis, pela confiança e incentivo dado ao longo de toda a minha formação académica.

Ao meu coorientador, Professor Doutor Michel Ferreira, os meus agradecimentos por me ter providenciado a oportunidade de desenvolver este projeto, agradecendo de igual modo ao Professor Doutor Luís Damas.

Agradeço ao meu orientador, Dr. José Alves, pela disponibilidade, apoio e profissionalismo que sempre teve comigo, não descartando as sugestões e comentários que me ajudaram a desenvolver este projeto.

A todos os meus colegas na Geolink, o meu muito obrigado por todo o conhecimento partilhado comigo e pelo apoio que sempre me facultaram.

RESUMO

O crescente ritmo tecnológico que se evidencia no nosso dia-a-dia, torna-se também evidente no mundo empresarial, com as empresas a quererem modernizar-se com vista a cativar mais clientes e satisfazê-los cada vez mais. Contudo a razão da modernização dos seus sistemas e serviços passa também por questões financeiras, pois em muitos casos, as empresas que escolham esse caminho, irão tirar partido de uma redução dos seus encargos financeiros. Esta redução é atingida através do aumento de produção, através da utilização de menores quantidades de matérias-primas ou a partir de um melhor e rápido sistema de atendimento dos seus clientes. Esta situação, também se irá verificar num *call center* caso se recorra a um sistema *Interactive Voice Response (IVR)*, para que um operador virtual realize algumas das funções que um operador dito humano efetue. Isto permite que este último fique com mais tempo para lidar com situações complexas que alguns clientes possam ter e que o operador virtual não possa resolver.

O presente relatório tem a função de apresentar o trabalho desenvolvido no âmbito da unidade curricular de Estágio, realizado na empresa Geolink. O objetivo deste trabalho é a conceção de uma operadora virtual, designada de “Josefina”, cujas capacidades serão idênticas às dos operadores humanos, libertando a carga de chamadas que estes têm diariamente de lidar.

O projeto apresentado surge como uma parte integrante de um *call center* desenvolvido pela Geolink, que se encontra atualmente em funcionamento em diferentes centrais de táxis em Portugal.

Neste relatório vão ser analisados alguns sistemas que disponibilizam um IVR como parte integrante de um *call center*, como forma de se perceber o conceito presente nesta tecnologia, sendo ainda explicadas as funcionalidades do IVR desenvolvido. Numa fase posterior é descrita a arquitetura da aplicação, a sua implementação e quais as ferramentas utilizadas no desenvolvimento do projeto aqui apresentado, fazendo por fim uma análise aos resultados da sua implementação.

Palavras-Chaves: Asterisk, *Call Center*, IVR, Operador(a).

ABSTRACT

The growing technological pace that is evident every day, it also becomes evident in the business world, with the companies wanting to modernize in order to win more customers and satisfy them more and more. However the reason for the modernization of their systems and services also involves financial matters, because in many cases, companies that choose this path will take advantage of a reduction in the financial burden. This decrease is achieved by increasing production through the use of smaller quantities of raw materials or from a better and fast service to their customers.

The situation set out above, it also being checked if a call center uses an Interactive Voice Response (IVR) system, so that a virtual operator can perform some of the functions that a human operator would do. This allows the human operator to have more time to deal with complex situations that some customers may have and the virtual operator can't resolve himself.

This report has the task of presenting the work undertaken within the course of Estágio and was conducted in the Geolink company. The aim of this work is the design of a virtual operator, called "Josefina", whose skills are identical to the skills that a human operator has, releasing the load of calls they have to deal with every day.

The project presented emerges as an integral part of a call center developed by Geolink, which is currently in operation in different central taxis in Portugal.

In this report will be analyzed some systems that provide an IVR as part of a call center, in order to realize the concept related with this technology, will be also explained the features of the IVR developed. Later in is described the application architecture, implementation, and which tools used in the development of the project presented here, at last an analysis of the results of this implementation, will be made.

Keywords: Asterisk, Call Center, IVR, Operator.

ÍNDICE

AGRADECIMENTOS	3
RESUMO	4
ABSTRACT	5
ÍNDICE DE FIGURAS	8
CAPÍTULO 1: INTRODUÇÃO	10
1.1. MOTIVAÇÃO	10
1.2. O PROJETO	14
1.2.1. A EMPRESA	14
1.3. ESTRUTURA	14
CAPÍTULO 2: ESTADO DE ARTE	17
2.1. CALL CENTER	17
2.2. INTERACTIVE VOICE RESPONSE	17
2.3. IVR PARA CALL CENTER	18
CAPÍTULO 3: REQUISITOS DO IVR CRIADO	21
3.1. FUNCIONALIDADES	21
3.1.1. NOTIFICAÇÃO DE CANCELAMENTO DO SERVIÇO	21
3.1.2. NOTIFICAÇÃO DE ATRIBUIÇÃO DE TÁXI	22
3.1.3. NOTIFICAÇÃO DE SUBSTITUIÇÃO DE TÁXI	22
3.1.4. NOTIFICAÇÃO DOS OPERADORES	22
CAPÍTULO 4: ARQUITETURA	25
4.1. DIAGRAMA DE COMPONENTES	25

4.2. FLUXO DE MENSAGENS	26
4.3. INTERVENÇÃO DO OPERADOR VIRTUAL	27
4.3.1. CANCELAMENTO DO SERVIÇO	28
4.3.2. NOTIFICAÇÃO DO CLIENTE	28
4.3.2.1. ATRIBUIÇÃO DE TÁXI	28
4.3.2.2. SUBSTITUIÇÃO DE TÁXI	29
4.3.3. NOTIFICAÇÃO DO OPERADOR	29
CAPÍTULO 5: IMPLEMENTAÇÃO	31
5.1. FERRAMENTAS UTILIZADAS	31
5.1.1. ASTERISK	31
5.2. CONFIGURAÇÃO DO ASTERISK	31
5.2.1. <i>QUEUES</i>	32
5.2.2. <i>PEERS</i>	32
5.2.3. <i>DIALPLANS</i>	33
5.2.4. <i>DIALPLANS DO ASTERISK</i>	34
5.2.5. CONFIGURAÇÕES DO IVR IMPLEMENTADO	35
5.3. IMPLEMENTAÇÃO DA OPERADORA VIRTUAL	37
5.3.1. <i>ROBOTOPERATOR</i>	37
5.3.2. <i>CALLOPERATOR</i>	38
5.3.3. <i>DIALPLAN</i>	39
CAPÍTULO 6: RESULTADOS	41
CAPÍTULO 7: CONCLUSÃO	45
ACRÓNIMOS	46
REFERÊNCIAS	47

ÍNDICE DE FIGURAS

Figura 1: Chamadas perdidas pelo <i>call center</i> da Raditáxis entre janeiro e fevereiro de 2014.....	11
Figura 2: Chamadas perdidas pelo <i>call center</i> da Raditáxis entre março e maio de 2014.....	11
Figura 3: Média de chamadas perdidas por hora de janeiro a maio de 2014	13
Figura 4: Terminal de operador para call center, desenvolvido pela Geolink.....	23
Figura 5: Diagrama da arquitetura do projeto	25
Figura 6: Diagrama do fluxo de mensagens	27
Figura 7: Número de chamadas recebidas por mês entre janeiro e maio de 2014.....	42
Figura 8: Média da probabilidade de <i>blocking</i> por mês entre janeiro e maio de 2014..	42

CAPÍTULO 1: INTRODUÇÃO

CAPÍTULO 1: INTRODUÇÃO

O presente relatório tem a função de apresentar o projeto “*Interactive Voice Response* para *Call Center* de Táxis” desenvolvido na empresa Geolink Lda, de novembro de 2013 a junho de 2014. O projeto foi proposto pela empresa no âmbito da unidade curricular de Estágio, do quinto ano do plano de estudos do curso de Mestrado Integrado em Engenharia de Redes e Sistemas Informáticos, da Faculdade de Ciências da Universidade do Porto.

Neste capítulo descreve-se o objetivo do trabalho realizado e o seu enquadramento na empresa e, por fim, descreve-se de forma sucinta os próximos capítulos.

1.1. Motivação

O projeto apresentado neste relatório tem como intuito desenvolver um produto que irá otimizar o serviço prestado por uma central de táxis, de forma a realizar os pedidos dos seus clientes de uma forma mais satisfatória, daquela verificada até à data.

Uma situação bastante preocupante, com o qual uma central de táxis se depara, é claramente o elevado número de chamadas com que lida diariamente, sendo que muitas chamadas não são atendidas. A partir de dados recolhidos junto da central de táxis, Raditáxis, foram perdidas em média 6% das chamadas diárias, entre janeiro e maio de 2014, tal como se pode observar nas figuras 1 e 2.

Pela análise dos dados apresentados é possível verificar que, em certas alturas do ano, a percentagem das chamadas perdidas sobe drasticamente. Essas situações acontecem em alturas críticas, sendo que na zona do Porto estão associadas, por exemplo, à passagem de ano ou a festas da cidade, onde o *call center* pode perder até 72% das chamadas que recebe.

Uma forma de combater estes números e aumentar a performance do *call center*, passa pela implementação do IVR desenvolvido neste projeto, uma vez, que os operadores podem ficar mais aliviados de certas funções. Assim podem atender um maior número de clientes, reduzindo a quantidade de chamadas perdidas do *call center*.

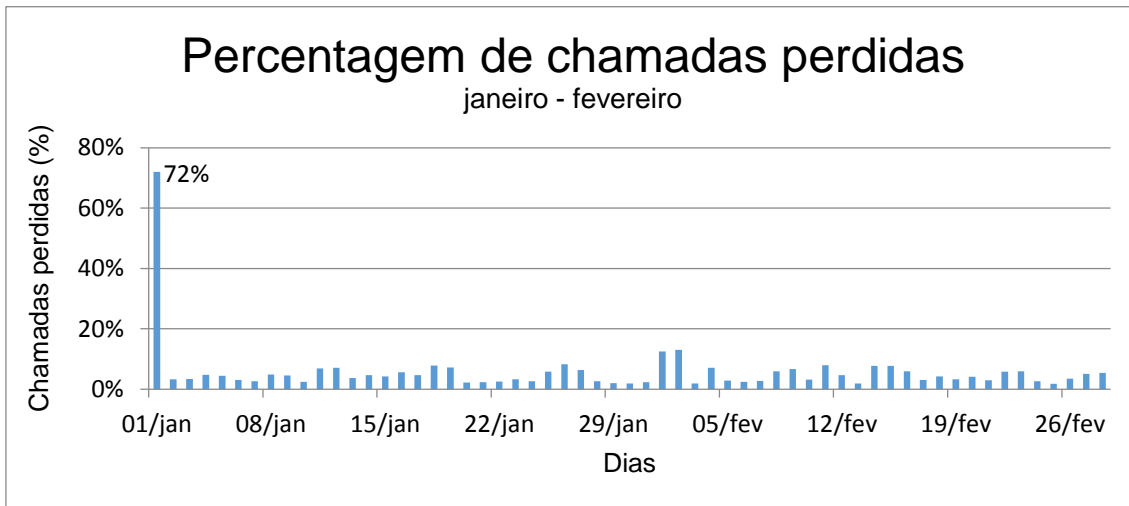


Figura 1: Chamadas perdidas pelo call center da Raditáxis entre janeiro e fevereiro de 2014

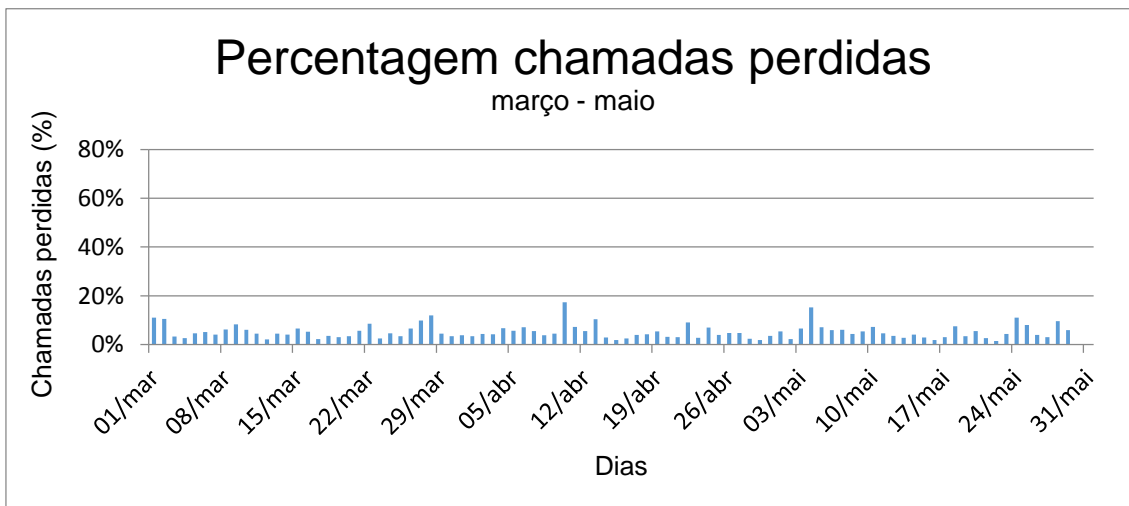


Figura 2: Chamadas perdidas pelo call center da Raditáxis entre março e maio de 2014

Na figura 3, encontra-se representada a média de chamadas perdidas por hora no *call center* Raditáxis, onde é possível concluir que o número de chamadas perdidas varia de acordo com o dia da semana, isto porque a afluência ao *call center* varia também de acordo com esse parâmetro. Assim, é possível analisar, por exemplo, que no período das 0h às 8h, o número de chamadas perdidas é acentuado na quarta-feira, onde se perderam, em média, 40 de 74 chamadas recebidas nesse período. Estes valores devem-se ao facto de o Ano Novo de 2014 ter acontecido numa quarta-feira, daí este elevado número de perda de chamadas.

Outro caso que requer uma análise é o dia da semana “domingo” entre as 0h e as 3h, ou seja, no primeiro turno do *call center*, onde se verifica, em média, uma perda de 13 chamadas em 140 recebidas.

É importante referir que o IVR será utilizado em certos períodos do dia, de acordo com a afluência de chamadas que o *call center* esta a receber, ou seja, quando o número de chamadas perdidas começa a ser elevado, o operador virtual é solicitado a intervir de forma a realizar certas funções.

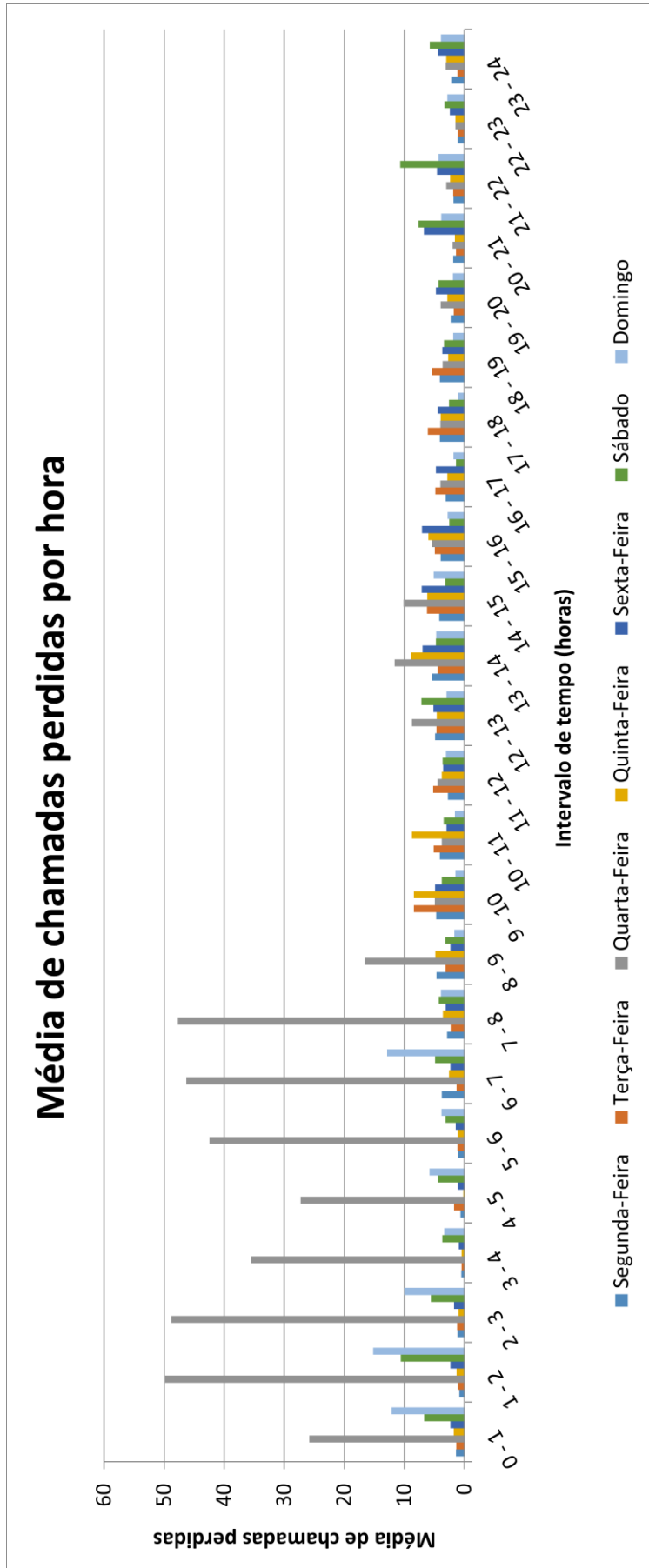


Figura 3: Média de chamadas perdidas por hora de janeiro a maio de 2014

1.2. O projeto

O projeto desenvolvido consistiu na criação de uma operadora virtual para atendimento robotizado por IVR para uma central de táxis, ou seja, foi desenvolvida uma operadora robotizada, para um *call center*, que tem a capacidade de realizar algumas das funções dos operadores humanos. Deste modo, foi utilizado um sistema, desenvolvido pela Geolink, onde se encontram todos os serviços de táxi, isto é, um sistema no qual é possível observar todos os pedidos solicitados pelos clientes.

A operadora “Josefina” fica encarregue de avisar um passageiro quando o seu serviço for cancelado, dando a este um conjunto de opções que pode escolher. Além disso, a operadora fica também encarregue de avisar o cliente caso seja trocado o número do táxi associado ao seu pedido, ficando ainda responsável pela divulgação do número do táxi que ficou atribuído, entre outras funcionalidades.

1.2.1. A empresa

A Geolink é uma empresa de base tecnológica, criada em dezembro de 2007, a partir de um *spin-off* do conhecimento desenvolvido na Faculdade de Ciências da Universidade do Porto (FCUP), na área de gestão de informação geoespacial.

A Geolink especializou-se na otimização de frotas automóveis, envolvendo dispositivos móveis que comunicam, em tempo real, com servidores centrais, que têm a função de monitorizar e otimizar o funcionamento dos veículos em tempo real. Os *call centers* ajudam na gestão das frotas tornando-se por esse motivo num componente fundamental das arquiteturas tecnológicas desenvolvidas pela Geolink [1].

1.3. Estrutura

No capítulo 2 é apresentado o estado de arte do projeto. Posteriormente no capítulo 3 são descritos os requisitos que o IVR a criar irá ter, dando foco à explicação das suas funcionalidades, isto é, quais são as funções que a operadora virtual é responsável por resolver.

No capítulo 4 é exposta a arquitetura do IVR criado, explicando-se com detalhe as suas componentes, sendo ainda, ilustrado como se propagam os principais sinais e mensagens entre os diferentes componentes deste sistema criado.

No capítulo 5 é clarificado o desenvolvimento e a implementação do IVR criado, referindo-se quais as ferramentas utilizadas e explicando-se com detalhe a estrutura e as abordagens escolhidas no projeto.

Por último, no capítulo 6 é explicado qual o impacto que a criação da operadora virtual teve no quotidiano dos *call centers* onde foi introduzido.

CAPÍTULO 2: ESTADO DE ARTE

CAPÍTULO 2: ESTADO DE ARTE

Com o desenvolvimento de novas tecnologias, que se torna visível no dia-a-dia, também os *call centers* tendem a melhorar o seu serviço, uma vez que dispõem de “novas armas”. Uma dessas “armas” é claramente o *interactive voice response* que se pretende criar no desenvolvimento deste projeto.

Este capítulo descreve de modo sucinto o funcionamento de um *call center* associado a uma frota de táxis. Explica ainda o que é e como funciona um sistema IVR de um modo geral, e quando este se encontra associado a um *call center*.

2.1. Call Center

A criação de *call centers* responsáveis pelo atendimento de chamadas de clientes, passou pela necessidade dos taxistas se associarem em frotas, uma vez que nas grandes cidades existe um grande número de táxis.

Assim as chamadas vindas de clientes deixam de ser atendidas pelos motoristas e passam a cair numa central de atendimento (*call center*) que recebe toda a informação do cliente. De seguida é contactado o veículo disponível mais próximo. Desta forma os clientes passam a necessitar apenas dos contactos da central, ao invés do contacto de diferentes motoristas.

Contudo, este método também tem as suas limitações: num período em que o fluxo de chamadas seja muito elevado, os operadores não podem garantir fluidez no atendimento nem podem garantir que todos os clientes em espera sejam atendidos. Isto acontece porque existe um número limitado de linhas e operadores, caso o número de chamadas, num determinado período, seja superior ao número de linhas/operadores, então as chamadas são rejeitadas.

2.2. Interactive voice response

O IVR consiste numa tecnologia que permite que um computador tenha a capacidade de interagir com seres humanos, sendo esta situação garantida através da interpretação de voz e/ou através da análise de sons que os teclados dos telefones e telemóveis produzem, denominados de sinais *Dual-Tone Multi-Frequency* (DTMF) [2].

Desta forma um indivíduo que interaja com o computador só tem de seguir as instruções deste e determinar quais são as condições que melhor satisfazem aquilo que se pretende.

Torna-se difícil atribuir o crédito da invenção deste sistema unicamente a um indivíduo, uma vez que vários deram o seu contributo, para que nos dias de hoje tivéssemos o sistema tal e qual como o conhecemos. Em 1937, Homer Dudley dos Laboratórios Bell criou um dispositivo denominado “the Voder”, que consistiu na primeira tentativa de analisar eletronicamente a voz humana, tendo sido desenvolvida ao longo dos anos [3].

Porém o IVR não podia apenas depender do sistema de reconhecimento de voz, e foi então que no ano de 1961, a Bell System desenvolveu o sistema DTMF, no qual cada tecla do telefone produz um som específico [4].

Nos anos 70, as empresas começaram a aderir em massa a esta tecnologia, apesar de esta ainda ser bastante dispendiosa e precoce, contudo já era possível uma interação de curta duração entre os seres humanos e o *robot* que se encontrava do outro lado da linha telefónica.

Quando os sistemas de *call center* se começaram a modernizar de forma significativa o IVR começou a crescer de uma forma mais notória, devido às características que os computadores e a própria tecnologia começaram a possuir.

Hoje em dia, o sistema IVR é tipicamente utilizado por serviços que recebem diariamente um elevado número de chamadas como forma de satisfazer aceitavelmente todas essas chamadas. Este sistema pode satisfazer as necessidades de uma parte dos clientes, deixando os operadores livres para os clientes que têm problemas mais específicos que o operador virtual não consegue solucionar.

Outra utilidade passa pela possibilidade de estender a sua operação 24 horas por dia, algo que com um operador humano é extremamente difícil de fazer, tornando-se bastante dispendioso.

2.3. IVR para *Call Center*

No nosso dia-a-dia, deparamo-nos muitas vezes com a interação com operadores virtuais, desde telefonar para solucionar um problema com a nossa ligação à Internet,

ligar para o banco ou até mesmo para a nossa operadora de telefone. Por esse motivo, existe um inúmero leque de serviços que utilizam este tipo de *software* (IVR).

Por não ser possível apresentar uma lista de todos esses serviços, apresenta-se de seguida uma listagem resumida: *inContact*, *Symtellelect* e *talkdesk*.

inContact. A *inContact* fornece um *software* IVR para *call center*, que segundo os seus criadores, “reduz os custos por chamada permitindo aos seus clientes escolher qual o tipo de ajuda que necessitam”. Para que se verifique essa situação, este *software* permite ao cliente resolver certas situações sozinho, sendo que as situações mais complicadas ficam à responsabilidade dos operadores do *call center* que este contacta [5].

Symtellelect. Consiste num serviço desenvolvido pela Enghouse Interactive, cuja política é “Help Your Customers Help Themselves” (ajude os seus clientes a ajudarem-se a si mesmos). Para tal, este serviço fornece, entre muitas funcionalidades, um IVR, que tem o objetivo de libertar os operadores de tarefas que os próprios clientes podem resolver [6].

talkdest. Este sistema consiste num *software* para *call center*, no qual esta empresa fornece um *interactive voice response*, que segundo os seus criadores tem quatro características principais: mensagens facilmente configuráveis, interações automáticas, roteamento inteligente e capacidade de lidar com grande número de chamadas [7].

Todos os serviços anteriormente listados, fornecem um IVR como componente dos seus softwares para *call center*. Apesar de cada um deles ter as suas próprias características, todos eles, têm uma característica em comum, tornar o atendimento dos clientes melhor, mais rápido e mais eficiente, sendo por esse motivo que disponibilizam um IVR para interagir com os seus clientes em primeiro lugar.

O IVR a ser criado no decorrer deste projeto irá ter características semelhantes aos dos enumerados acima. A Geolink propôs a sua criação, como forma de satisfazer os seus clientes de forma rápida e eficiente, evitando que muitas das chamadas não sejam perdidas principalmente em dias em que a afluência aos táxis é muito acentuada.

CAPÍTULO 3: Requisitos do IVR criado

CAPÍTULO 3: REQUISITOS DO IVR CRIADO

Este capítulo descreve os requisitos do IVR criado dando foco às suas funcionalidades, fazendo uma descrição sucinta das mesmas.

3.1. Funcionalidades

Aquando da apresentação deste trabalho, a empresa decidiu quais seriam as principais funcionalidades deste projeto, isto é, quais as funções que a operadora virtual “Josefina” é responsável por efetuar.

As funções a desempenhar serão as seguintes:

- Notificar o cliente caso o seu pedido seja cancelado;
- Notificar o cliente do número de táxi atribuído ao seu pedido;
- Notificar o cliente caso haja uma substituição de táxi do pedido;
- Notificar os operadores de que existem pedidos que necessitam de intervenção humana.

Todas as funcionalidades anteriormente referidas são possíveis de ativar e desativar em qualquer situação, utilizando para esse efeito um IVR auxiliar onde cada uma das funcionalidades tem as opções de ativar e desativar. Desta forma o *call center* pode determinar quais as funções de que a operadora virtual irá ser responsável.

As funcionalidades acima referidas permitem automatizar as operações que os operadores efetuam. Como consequência os operadores irão poupar tempo, podendo realizar outro tipo de operações no *call center*.

Descreve-se de seguida cada uma das funcionalidades referidas anteriormente.

3.1.1. Notificação de cancelamento do serviço

O cancelamento do serviço associado a um cliente pode ocorrer devido a inúmeras situações, por exemplo, o operador pode atribuir o serviço a um táxi e posteriormente pode ter a necessidade de cancelá-lo devido a esse táxi não se encontrar disponível

no momento em causa, sendo necessário avisar o cliente dessa situação. Até o IVR surgir teria de ser o operador a fazer esse trabalho, perdendo tempo útil em que poderia estar atender novos pedidos de clientes.

Com o IVR, essa função fica da responsabilidade da operadora virtual “Josefina”. Quando o operador realiza o cancelamento do serviço, o servidor recebe um sinal com o CLI (*Caller Line Identification*) do cliente do pedido que foi cancelado, sendo que nesse momento o cliente é contactado.

Depois de informado, é-lhe apresentado um conjunto de opções: cancelar o seu pedido, aguardar que lhe seja atribuído outro táxi, ou então falar com o operador.

3.1.2. Notificação de atribuição de táxi

A necessidade de notificar o cliente quando um táxi é atribuído ao seu pedido pode ocorrer em duas situações distintas. A primeira ocorre quando o operador desliga a chamada ao cliente e só depois é que atribui o táxi ao pedido, sendo o cliente posteriormente avisado de qual o número do seu táxi. Outra situação ocorre quando o cliente, após ser informado que o seu pedido foi cancelado decide aguardar que lhe seja atribuído um novo táxi. É então nestes casos que a operadora “Josefina” tem a função de ligar ao cliente a informar a atribuição.

3.1.3. Notificação de substituição de táxi

Outra ocorrência em que é necessário informar o cliente sobre o estado do seu pedido, acontece quando por algum motivo o táxi cancela o pedido (devido a avaria, trânsito, entre outros). Assim, caso o táxi seja substituído é necessário que a operadora virtual comunique ao cliente o número do novo táxi.

3.1.4. Notificação dos operadores

Este tipo de notificação é exclusiva para o *call center* e tem a função de avisar o operador que tem serviços que requerem a sua intervenção, ou seja, serve para avisar o operador, quando este não se encontra no terminal de operador (figura 4), desenvolvido pela Geolink, que tem serviços pendentes e que tem de os resolver.

Este tipo de notificação foi solicitado por um *call center* em específico, uma vez que este não dispõe de operadores no turno da noite, portanto é necessário avisar que existem serviços pendentes, nesse turno.

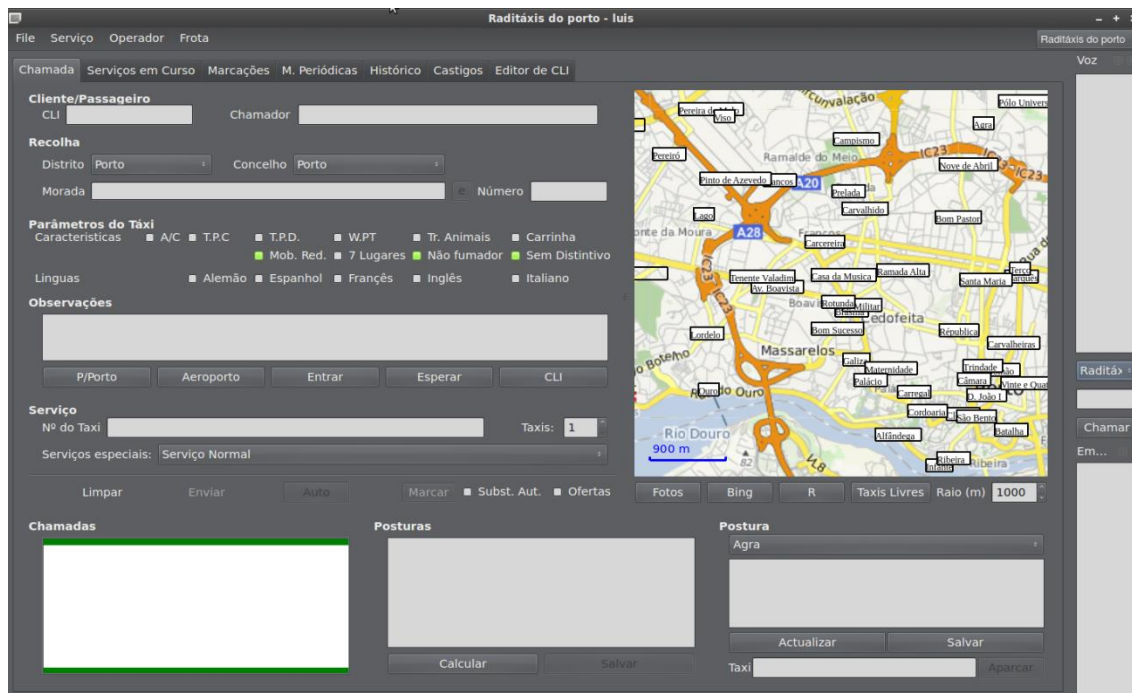


Figura 4: Terminal de operador para call center, desenvolvido pela Geolink

CAPÍTULO 4: Arquitetura

CAPÍTULO 4: ARQUITETURA

No presente capítulo é apresentada a arquitetura do IVR criado, identificando e descrevendo todas as suas componentes. Posteriormente, é clarificado como se propagam as mensagens entre os diferentes componentes do sistema, referindo qual a importância de tais mensagens.

4.1. Diagrama de componentes

A figura 5 representa a arquitetura do sistema IVR criado, onde se representa os seus diferentes componentes, bem como as ligações que estes têm entre si. Dos componentes apresentados, os que se encontram no fundo cinzento consistem naqueles que foram implementados de raiz, sendo que os restantes componentes do sistema foram previamente desenvolvidos pela Geolink.

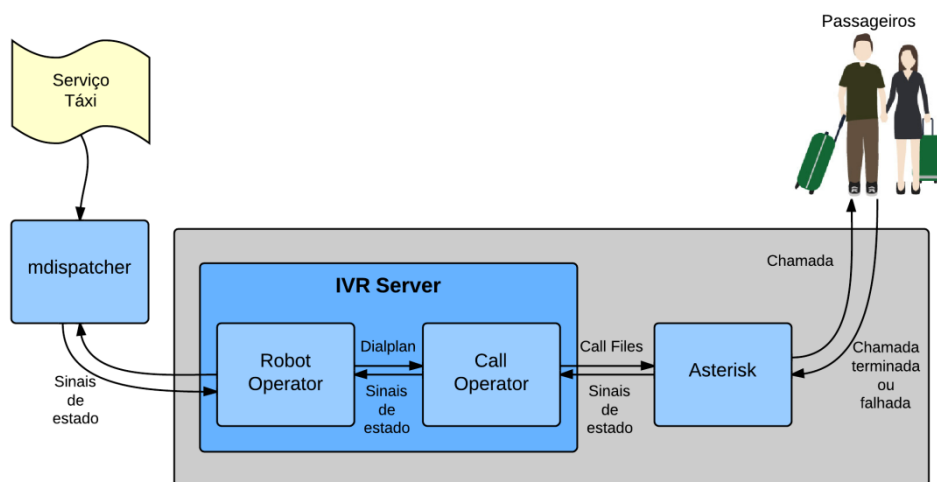


Figura 5: Diagrama da arquitetura do projeto

De seguida, apresenta-se uma descrição sobre as funcionalidades de cada uma das componentes, representados na figura anterior.

mdispatcher. Consiste num módulo do servidor de despachos de táxis da Geolink, responsável pela monitorização dos serviços de táxi que se encontram

disponíveis e pela emissão de sinais de acordo com a circunstância em que os serviços se encontram, como por exemplo, serviço atribuído, serviço cancelado, entre outros.

Robot Operator. Constitui uma das duas partes que formam o IVR Server, onde a sua função é a análise dos sinais que recebe do *mdispatcher*, tratando-os de acordo com as suas características. Aquando do tratamento dos eventos, o *Robot Operator* tem de comunicar com o *Call Operator*, enviando-lhe *dialplans* de acordo com os eventos que recebe anteriormente.

Call Operator. Este componente do IVR Server recebe *dialplans* do *Robot Operator*, criando e enviando *call files* para o Asterisk - estes *call files* são criados de acordo com a informação presente nos *dialplans*. O *Call Operator* fica posteriormente à espera que o Asterisk o notifique sobre o que se passou com a chamada gerada a partir do *call file*. Os *dialplans* que são recebidos pelo *Call Operator* são tratados sequencialmente.

Asterisk. Este serviço é responsável por realizar chamadas para os Passageiros, de acordo com os *call files* recebidos. Após a realização da chamada, o Asterisk recebe informações sobre a mesma, notificando de seguida o *Call Operator*.

4.2. Fluxo de mensagens

Nesta secção é explicado o fluxo de mensagens associado a cada um dos eventos que são importantes para este projeto. As explicações vão ser dadas tendo em conta a figura 6, onde se encontra representado o diagrama do fluxo de mensagens genérico.

Consoante os componentes envolvidas, existem diferentes tipos de trocas de mensagens, tal como se observa na figura 5, sendo estes explicados de seguida.

Sinais de estado. Neste tipo de fluxo de mensagens, aquilo que se transmite de um componente para outro, corresponde a um sinal que notifica o seu recetor sobre aquilo que se passa numa determinada situação, ou seja, caso um serviço de táxi seja cancelado, o recetor dessa mensagem recebe um sinal que o informa da ocorrência dessa situação.

Dialplans. Neste caso, a troca de mensagens, consiste em propagar um *dialplan* de um componente para outro. Conforme a situação que se pretenda resolver, existem

diferentes *dialplans*, ou seja, para cada uma das funções da operadora virtual existe um determinado *dialplan*.

Chamada. Este tipo consiste na realização de uma chamada para um determinado passageiro, sendo que apenas se verifica entre o Asterisk e o cliente em causa, uma vez que é o Asterisk o responsável pela realização das chamadas telefónicas, tal como já foi anteriormente referido.

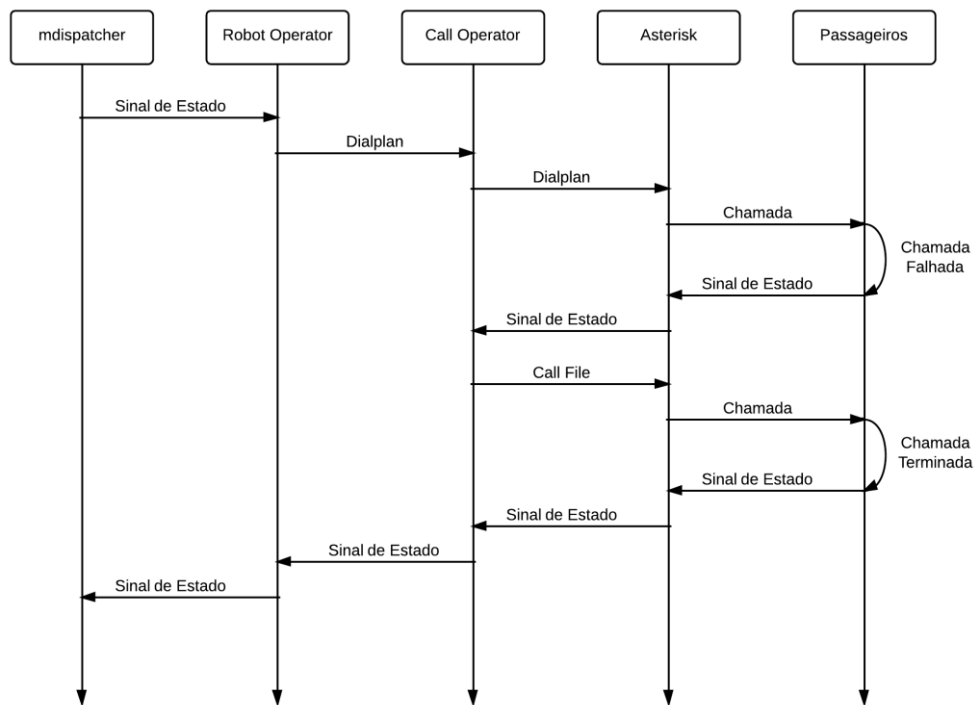


Figura 6: Diagrama do fluxo de mensagens

4.3. Intervenção do operador virtual

Procede-se de seguida à explicação das situações em que o operador virtual é responsável por intervir, dando ênfase, à comunicação dos diferentes componentes deste projeto.

4.3.1. Cancelamento do serviço

Caso o *Robot Operator* receba um evento que o notifica do cancelamento de um dado serviço, este vai gerar um *dialplan* de acordo com esta situação, notificando de seguida o *Call Operator*.

O *Call Operator*, como já foi anteriormente referido, vai ter a responsabilidade de gerar um ficheiro *call file*, passando-o ao Asterisk, de maneira a que seja feita uma chamada para avisar o passageiro da ocorrência dessa situação.

Contudo, várias circunstâncias podem ocorrer a essa chamada. Caso a chamada seja rejeitada ou falhe é necessário voltar a entrar em contacto com o passageiro, sendo enviado um sinal ao *Call Operator* para que volte a ligar ao cliente. Se a chamada for concluída com sucesso é necessário receber essa informação, isto porque, neste caso o passageiro terá de escolher uma hipótese daquelas que a operadora virtual lhe transmite. O sinal a ser recebido pelo *Call Operator* vai depender da escolha que o cliente realizou.

4.3.2. Notificação do cliente

Existem dois motivos pelos quais é necessário notificar o cliente, no que diz respeito ao táxi associado ao seu pedido. Esses motivos encontram-se enumerados e explicados de seguida.

4.3.2.1. Atribuição de táxi

Esta situação ocorre quando o *mdispatcher* envia um sinal ao *Robot Operator* informando que foi atribuído um táxi a um dado pedido. Posto isto, este último, vai numa primeira fase verificar se o operador se encontra em linha com o passageiro. Em caso negativo vai ser criado um *dialplan*, sendo posteriormente passado ao *Call Operator*.

O *Call Operator* envia um *call file* para o Asterisk, que efetua a chamada para o passageiro, transmitindo de seguida um sinal ao *Call Operator* para informá-lo sobre o sucesso ou insucesso da chamada efetuada. Em caso de insucesso a chamada é novamente realizada.

4.3.2.2. Substituição de táxi

Nesta situação o procedimento é idêntico ao da situação anterior, sendo que nesta a única diferença passa por não se verificar se o cliente se encontra em linha ou não. Isso acontece porque, neste caso, o cliente vai ser notificado depois de ter sido contactado aquando de o seu serviço ter sido cancelado, e este escolheu aguardar que um novo táxi lhe fosse atribuído. Sendo assim é lhe transmitida a informação e o Asterisk informa o *Call Operator* sobre o sucesso ou insucesso da chamada efetuada. Mais uma vez, em caso de insucesso a chamada é novamente efetuada.

4.3.3. Notificação do operador

Caso o *Robot Operator* receba um sinal que o informa de que existem serviços no terminal de operador, que necessitam de ser tratados e caso esta notificação esteja ativa, então é criado um *dialplan*, que é passado numa fase posterior para o *Call Operator*.

O *Call Operator* recebe esse *dialplan* e envia um *call file* para o Asterisk, ficando este responsável por contactar o operador. Este último tem ainda a função de transmitir ao *Call Operator* o sucesso ou insucesso da chamada, sendo que em caso de insucesso é contactado outro operador presente num *array* de contactos, existente nesta componente.

CAPÍTULO 5: Implementação

CAPÍTULO 5: IMPLEMENTAÇÃO

O presente capítulo tem como objetivo clarificar o desenvolvimento e implementação do IVR criado. Numa primeira fase são referidas quais as ferramentas utilizadas para se atingir o projeto final, e em seguida, são explicados os detalhes da implementação da operadora virtual, dando ênfase à estrutura e às abordagens escolhidas ao longo do desenvolvimento deste projeto.

5.1. Ferramentas utilizadas

5.1.1. Asterisk

O Asterisk foi criado em 1999, criando-se a primeira plataforma *open source* de telefonia. Este sistema foi criado para ser possível desenvolver de raiz uma aplicação de comunicação robusta, transformando um computador num servidor de comunicações. Consiste num *software private branch exchange* (PBX), que disponibiliza um conjunto de serviços, como *voicemail*, *call conferencing*, *interactive voice response*, *call queuing*, entre outros, que permitem a implementação de um sistema de telefonia [8][9].

A escolha da utilização deste sistema foi imposta pela Geolink devido às características acima referidas.

5.2. Configuração do Asterisk

Para que seja possível a operadora virtual notificar os clientes é necessário criar um conjunto de estruturas, como forma de esta ter a capacidade de comunicar com os clientes. Estas estruturas são as *queues*, os *peers*, os *dialplans* e os *dialplans* do Asterisk, para que a cada situação seja realizado um determinado procedimento.

As secções seguintes têm a função de clarificar qual a função e o funcionamento de cada uma das estruturas anteriormente referidas. Os exemplos apresentados nestas secções são apenas exemplos informativos, não constituindo a implementação usada no IVR criado neste projeto.

5.2.1. Queues

No `queues.conf` define-se quais são as *queues* de chamadas [10]. Este ficheiro serve para criar as *queues* onde as chamadas irão entrar. De seguida apresenta-se um exemplo de uma *queue*:

```
[callcenter]
strategy = rrmemory

member => SIP/user1
```

Neste exemplo foi criada uma *queue*, denominada *callcenter*, onde se define a sua estratégia e os seus membros.

A estratégia serve para informar o Asterisk de como lidar com as chamadas dividindo-as pelos operadores do *call center*. Para este caso, foi escolhida a *rrmemory*, que consiste na estratégia *round robin*, de modo a que quando uma chamada é recebida esta é passada por entre os operadores até que um atenda. Será também esta a estratégia a usar na implementação do IVR.

Os membros são definidos de forma a identificar quais são os operadores responsáveis por aquela *queue*, ou seja, quais os operadores que vão atender chamadas recebidas na *queue* em questão.

Procede-se de seguida à explicação de como são configurados esses membros.

5.2.2. Peers

Os *peers* são configurados no ficheiro `sip.conf`, onde se define quem tem a capacidade de receber e/ou fazer chamadas [11]. Apresenta-se de seguida um exemplo de uma configuração deste ficheiro:

```
[user1]
secret=pass
type=friend
```

No exemplo apresentado, foi criado um utilizador, *user1*, cuja *password* se encontra definida no parâmetro *secret*. O seu tipo é *friend*, o que significa que este utilizador pode fazer e receber chamadas.

Os *peers* são definidos através da tecnologia (*technology*) usada e da sua identificação (*id*) como se apresenta de seguida: `technology/id`. Na implementação deste projeto foi utilizada a tecnologia SIP (*Session Initiation Protocol*), pois é aquela que é utilizada pela Geolink em todos os seus sistemas de telefonia.

5.2.3. Dialplans

Os *dialplans* que serão criados, na componente *Call Operator*, têm a função de armazenar informação que será utilizada para gerar ficheiros do tipo *call files* (`.call`), que consiste num tipo de ficheiros do sistema Asterisk, responsáveis por gerar automaticamente chamadas para um determinado CLI utilizando este sistema de telefonia [12].

Os *call files* têm a sua sintaxe dividida em dois grandes grupos. O primeiro grupo diz respeito à parte onde a sintaxe informa como e para quem se deve fazer a chamada e o segundo grupo é utilizado para saber o que fazer quando a pessoa para quem se ligou atendeu a chamada realizada pelo Asterisk.

Para cada uma das funcionalidades referidas na secção 3.1, encontra-se associado um *dialplan*, ou seja, para cada uma das funções da operadora virtual irá ser criado um ficheiro *call file* que vai ser colocado no Asterisk (na pasta `var/spool/asterisk/outgoing/`), sendo que quando a chamada for efetuada este ficheiro deixa de existir.

Apesar de cada uma das funções da “Josefina” ter um *call file* diferente associado, todos têm a mesma estrutura, tal como se apresenta de seguida [12]:

- **Channel**, onde se refere qual o *channel* utilizado para fazer a chamada (ex: `SIP/xxxxxxxxx@x.x.x.x`);
- **Callerid**, corresponde, tal como o nome sugere, ao *id* do *caller*;
- **Context**, refere qual o *context* escolhido do ficheiro `extensions.lua`;
- **Extension**, refere qual a *extension* escolhida do ficheiro `extensions.lua`;
- **SetVars**, conjunto de variáveis que podem ser úteis.

Relativamente aos dois grupos referidos anteriormente, o Channel e o Callerid são do grupo que informa como e para quem se deve fazer a chamada e os restantes dizem

respeito ao grupo que é utilizado com o intuito de se saber o que fazer quando a pessoa para quem se ligou atendeu a chamada.

Em geral a estrutura de um *call file* não irá mudar muito, alterando-se o *context* e o *extension* escolhidos. O *channel*, vai mudar de acordo com o CLI fornecido e eventualmente as variáveis que se encontram no SetVars poderão também variar consoante a situação.

Contudo a mera criação destes ficheiros não garante que se realizem as chamadas telefónicas para os clientes, ou seja, não são suficientes para que a operadora virtual troque informações com os passageiros. Para isso é necessário criar os contextos e as extensões utilizadas pelo Asterisk cujos nomes se encontram presentes nos *dialplans* criados, como se encontra apresentado anteriormente.

5.2.4. Dialplans do Asterisk

Os *dialplans* do Asterisk definem o que vai acontecer quando uma chamada chega a este sistema. Para isso utilizam as estruturas denominadas de contextos onde são agrupadas várias extensões [13]. De seguida apresenta-se um exemplo de um *dialplan*:

```
[mycontext]
exten => 2000,1,Log(NOTICE, Entered hello)
exten => 2000,2,Answer()
exten => 2000,3,Say>Hello)
exten => 2000,4,Hangup()

exten => 4000,1,Log(NOTICE, Entered bye)
exten => 4000,2,Answer()
exten => 4000,3,Say>Bye)
exten => 4000,4,Hangup()
```

No *dialplan* apresentado está definido um contexto, *mycontext*, onde estão agrupadas duas extensões distintas. Na extensão 2000, assim que a chamada é recebida, é dado um *log* informativo de que isso se passou. De seguida a chamada é atendida, ouve-se uma voz na linha a dizer “*hello*” e a chamada é desligada por fim. Na outra extensão representada, o procedimento é mesmo, com a diferença que na linha ouve-se uma voz a dizer “*Bye*”.

O código anteriormente apresentado foi criado no ficheiro *extensions.conf*, que consiste no ficheiro que geralmente se utiliza para implementar este tipo de estruturas.

Contudo como este tipo de ficheiros são pouco flexíveis e difíceis de integrar com outras linguagens, nomeadamente Python, foi utilizado para a implementação deste projeto, o ficheiro `extensions.lua`, que torna essa integração muito mais facilitada.

De seguida apresenta-se o mesmo exemplo de *dialplan*, anteriormente mostrado, utilizando a linguagem LUA [14]:

```

extensions = {
  mycontext = {
    [2000] = exten1,
    [4000] = exten2,
  };
};

function exten1(c, e)
  app.Log("NOTICE", "Entered hello")
  app.Answer()
  app.Say("Hello")
  app.Hangup()
end

function exten2(c, e)
  app.Log("NOTICE", "Entered hello")
  app.Answer()
  app.Say("Bye")
  app.Hangup()
end

```

Como se pode observar o procedimento realizado neste exemplo é o mesmo que no ficheiro `extensions.conf`, apenas se alterando a estrutura do mesmo. Neste caso, primeiro definem-se os contextos e as extensões associadas a cada contexto, sendo que para cada uma das extensões existe uma função que é responsável pela implementação daquilo que se passa quando uma chamada é recebida na extensão em causa.

5.2.5. Configurações do IVR implementado

No caso do IVR implementado, cada extensão irá ser responsável por lidar com a chamada de acordo com o caso em que esta se encontra inserida, sendo que será nas extensões que a voz da operadora virtual irá ser carregada através de ficheiros áudio, como forma do cliente saber de que se trata a chamada recebida. É também aqui que se sabe o que o cliente pretende fazer quando lhe é disponibilizada mais do que uma

opção, como por exemplo, quando este recebe uma chamada a notificá-lo do cancelamento do seu pedido, em que se fornece mais do que uma opção ao passageiro para lidar com essa situação.

Tal como referido na secção 5.2.3, cada função da operadora irá ter um *call file* associado, do mesmo modo que cada uma dessas funcionalidades terá o seu contexto correspondente.

Em cada um desses contextos vamos ter uma extensão responsável por lidar com a chamada de acordo com as suas características, sendo que o nome dessa extensão se encontra representado no ficheiro *call file*. Em cada um dos casos a tratar temos ainda mais duas extensões além da já referida, uma para tratar dos casos em que a chamada falhou e outra para tratar dos casos em que o recetor da chamada desliga a mesma.

Para tratar do caso em que a chamada falhou, foi criada uma extensão do tipo *failed*, que corresponde a uma *standard extensions* do Asterisk [15], que é utilizada exclusivamente para tratar deste tipo de situações. No IVR implementado, sempre que uma chamada falha é necessário contactar novamente o cliente, sendo isso que se verifica quando a chamada “entra” na extensão *failed*.

Quando uma chamada é desligada pelo cliente, torna-se necessário tomar alguma medida relativamente a essa situação, que irá depender do motivo do contacto. Para esta situação é novamente utilizada uma extensão *standard* do Asterisk, denominada de extensão h (*hangup*) [15].

Como foi referido, dependendo do motivo do contacto, lida-se com o desligar da chamada de modo diferente. No caso de se contactar para notificar o cancelamento do serviço, caso o cliente decida desligar a chamada estamos perante a opção de cancelamento do seu pedido e como tal avisa-se o servidor dessa situação. Para as restantes situações o desligar da chamada é utilizado como forma de se ter a certeza que o cliente foi notificado relativamente à situação pelo qual foi contactado.

Apesar de já estarem criadas as duas estruturas anteriormente referidas, ainda se torna necessário desenvolver outras estruturas, as quais irão lidar com os eventos recebidos pelo servidor que contém toda a informação dos serviços de táxi que estão a decorrer. Estas estruturas são explicadas com detalhe na secção seguinte.

5.3. Implementação da operadora virtual

A estrutura pela qual o projeto se encontra organizado, foi proposta pela empresa Geolink, como forma de seguir a organização que esta tem em todos os seus projetos. Sendo assim, para o desenvolvimento da operadora virtual, foram criadas três classes, que se encontram descritas de seguida.

5.3.1. *RobotOperator*

A classe *RobotOperator* é utilizada para dois propósitos: monitorizar todos os eventos de todos os táxis recebidos do servidor e lidar com esses eventos.

Do *mdispatcher* são recebidos vários tipos de sinais existindo, por exemplo, para os seguintes casos: quando um serviço é aceite pelo táxi, quando temos o cancelamento do pedido pelo táxi, quando o operador cancela um pedido, entre outros.

Para a realização deste projeto as atenções vão se focar em três tipos de eventos, sendo eles:

- `service_canceled_by_op`;
- `waiting_op`;
- `service_assigned`.

O evento **`service_canceled_by_op`** ocorre quando o operador cancela um serviço que está a decorrer, sendo que essa situação pode acontecer devido a vários fatores, como por exemplo: cancelamento porque o cliente não apareceu ou cancelamento por pedido do cliente. Quando se está na presença de um evento deste tipo, é necessário notificar o cliente como forma de este ser informado que o seu pedido foi cancelado.

O **`waiting_op`** é um evento que ocorre quando existem serviços a decorrer no terminal que requerem a intervenção do operador. Quando esta notificação ocorre é necessário entrar em contacto com o operador em causa, de modo a que este tome as devidas medidas, para satisfazer os serviços que se encontram pendentes no terminal de operador. É importante referir que essa notificação só sucede quando o operador não se encontra no terminal.

Por último, o evento **`service_assigned`**, verifica-se quando o operador atribui a um dado pedido um determinado táxi, e como tal, é necessário em certas situações avisar

o cliente dessa atribuição. Essas situações passam pelos casos em que os clientes não se encontram “em linha” com o operador, ficando à responsabilidade da “Josefina” notificar o cliente da ocorrência dessa situação.

Independentemente da situação que ocorra, das anteriormente enumeradas, em todas elas se irá criar um *dialplan* que posteriormente será colocado numa *queue* que se encontra na classe *CallOperator*. Os *dialplan* criados, utilizando a classe *Dialplan*, irão ser diferentes de situação para situação, adaptando-se às circunstâncias da mesma.

5.3.2. *CallOperator*

Esta classe é utilizada para criar os ficheiros que são utilizados para a realização de chamadas e para garantir que os contactos a realizar são feitos por ordem de chegada dos mesmos. Para tal é utilizada uma *queue* de execução, onde são colocados os *dialplan's*, por ordem de chegada. Isto acontece porque temos uma arquitetura do tipo “*first in, first out*”, ou seja, o primeiro *dialplan* a ser colocado na fila de espera é o primeiro a ser tratado.

Como foi referido anteriormente, esta classe gera ficheiros, do tipo *call file* (.call). Estes ficheiros são gerados através do *dialplan* que esta classe recebe do *RobotOperator*, que apenas contém a informação que o ficheiro irá ter de acordo com a situação em que está inserido, ou seja, de acordo com o evento que o *RobotOperator* recebeu.

Depois da criação deste ficheiro, a *CallOperator*, tem ainda a responsabilidade de copiar este ficheiro para o sistema Asterisk, colocando num local onde estes são utilizados para automaticamente realizar chamadas.

Quando a chamada for efetuada, esta classe fica à espera da receção de um sinal que lhe transmita aquilo que se passou com a mesma, ou seja, a cada acontecimento que possa ocorrer com a chamada recebe-se um sinal que traduz essa situação. Por exemplo, caso o cliente não atenda a chamada, a classe *CallOperator* recebe um sinal que a informa da ocorrência dessa situação e, neste caso, o cliente irá ser contactado mais tarde.

Desta forma, caso a chamada falhe, seja desligada, ou realizada com sucesso, esta classe vai sempre receber um sinal informativo sobre a situação específica.

É importante referir que também são recebidos sinais quando o cliente faz uma escolha mediante aquilo que a operadora virtual lhe transmite, ou seja, na situação em

que este é avisado que o seu pedido foi cancelado, o cliente tem várias opções, sendo que a cada uma se encontra associado um sinal, para que a classe *CallOperator* possa determinar aquilo que tem de ser feito.

5.3.3. *Dialplan*

Esta classe é utilizada para criar os *dialplans* que a componente *Robot Operator* pretende. A sua estrutura é constituída por uma classe geral e por classes que estendem a classe geral.

Assim de acordo com o caso que o *Robot Operator* está a lidar, uma das classes auxiliares é usada estendendo a classe geral, onde se encontram definidos todos os parâmetros necessários para o *dialplan*. Na classe auxiliar são alterados os valores dos parâmetros de acordo com a situação que se está a lidar.

Por exemplo, caso o *Robot Operator* pretenda um *dialplan* para lidar com a situação de substituição de táxi, é utilizada a classe auxiliar *ChangeTaxiDialplan*, que vai estender a classe geral, alterando de seguida os valores dos contextos, das extensões e de variáveis importantes, como por exemplo, o CLI do cliente a contactar.

CAPÍTULO 6: RESULTADOS

CAPÍTULO 6: RESULTADOS

Com a criação do IVR pretendido neste projeto, o *call center* onde este se encontra inserido, dispõe de mais um operador, ou seja, a probabilidade de ser perderem chamadas vai diminuir. O operador virtual não vai ser responsável por lidar diretamente com os clientes, mas realiza funções que os operadores humanos realizavam, libertando estes dessas funções, para que lhes seja possível satisfazer um maior número de clientes.

A probabilidade anteriormente referida é calculada através da fórmula de Erlang B, que fornece como resultado a probabilidade de bloqueio (*blocking*) do *call center* em causa. De seguida apresenta-se a fórmula de Erlang B:

$$P_b = \frac{\frac{E^m}{m!}}{\sum_{i=0}^m \frac{E^i}{i!}}$$

Na fórmula apresentada a variável **m** corresponde ao número de operadores responsáveis pelo atendimento, a variável **E** consiste no produto do número de chamadas recebidas pelo tempo de conversação, sendo que este valor tem que estar uniformizado.

No gráfico da figura 8 é possível observar a média da probabilidade de *blocking* por mês entre janeiro e maio de 2014, na central de táxis, Raditáxis, onde é visível um decréscimo do valor dessa probabilidade. Esta redução deve-se ao decréscimo que também se evidencia no número de chamadas recebidas pelo *call center* (figura 7). Com a redução do número de chamadas a probabilidade de *blocking* também irá baixar.

Um fator que permite a probabilidade de *blocking* diminuir é como foi anteriormente referido, a redução de chamadas recebidas. Contudo um *call center*, como o da Raditáxis, não pretende essa situação, uma vez que menos chamadas significa menos clientes, portanto é necessário encontrar outro fator que realize o mesmo procedimento.

Outra forma de reduzir a probabilidade de *blocking* passa por aumentar a variável **m** da fórmula apresentada, ou seja, é necessário aumentar o número de operadores. Esta situação verifica-se, em parte, com a introdução do IVR criado neste projeto, uma vez que o operador virtual fica apenas responsável por certas funções, não lidando

diretamente com os pedidos dos clientes, tendo apenas a responsabilidade de libertar os restantes operadores de certas ocupações.

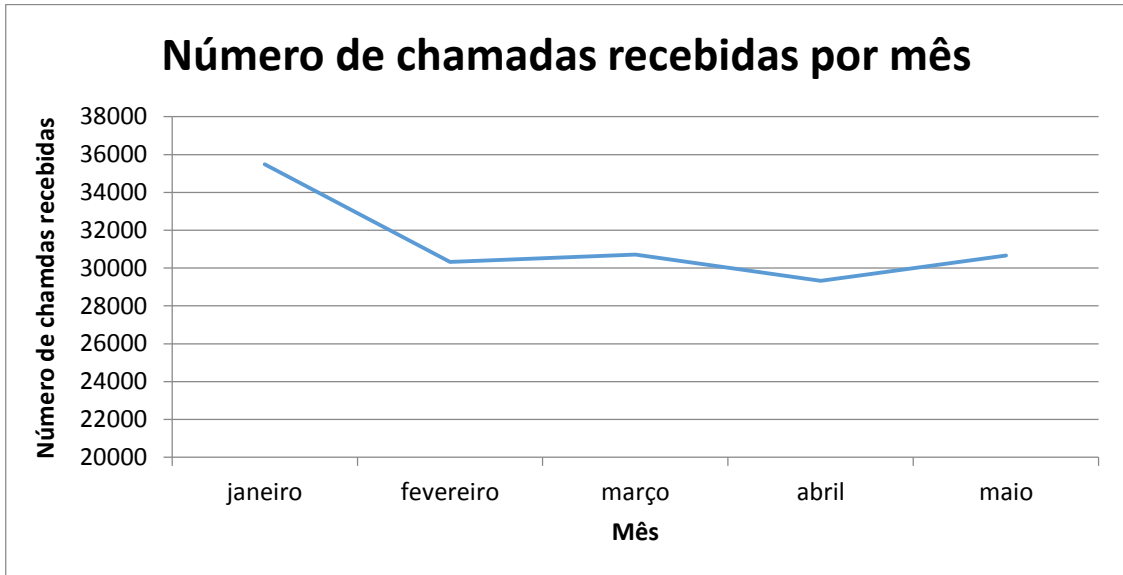


Figura 7: Número de chamadas recebidas por mês entre janeiro e maio de 2014

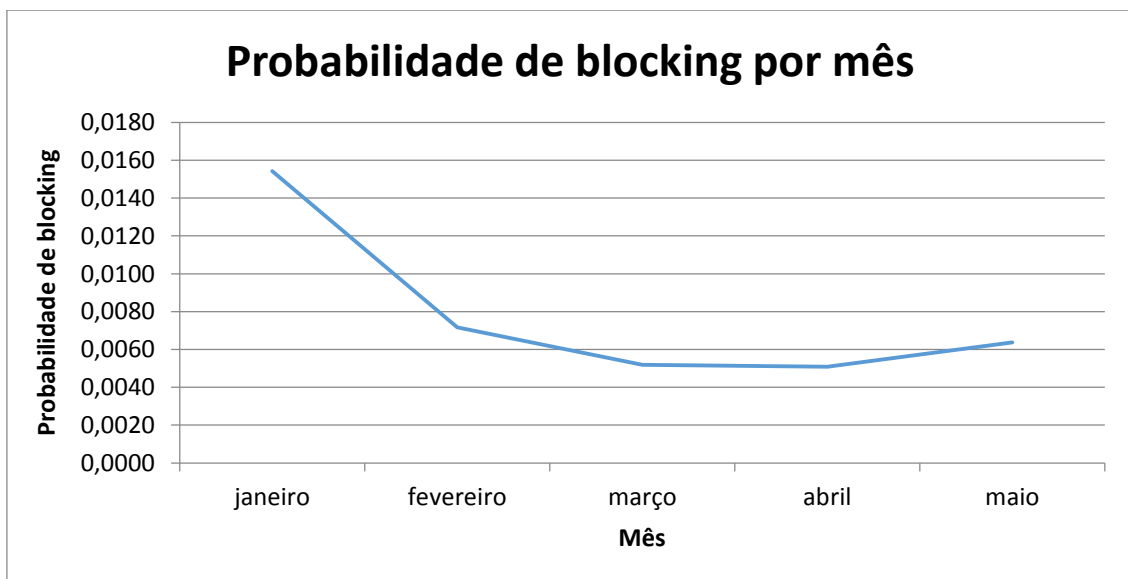


Figura 8: Média da probabilidade de *blocking* por mês entre janeiro e maio de 2014

Pela análise dos gráficos das figuras 3, 7 e 8 é possível concluir que a performance do *call center* da Raditáxis, antes da implementação do IVR, não é perfeita, pois em certos períodos de tempo torna-se necessário dispor de mais operadores para realizarem o atendimento dos clientes. Contudo, essa situação é impossível de satisfazer pelo que o recurso ao operador virtual irá melhorar significativamente a performance deste *call center*. Isto acontece porque certas funções (capítulo 3) deixam de ser realizadas pelos operadores humanos e passam a ser feitas pela “Josefina”, ficando estes com mais tempo para lidar com os pedidos dos clientes.

É importante referir que em certos períodos de tempo de um determinado dia, o número de chamadas perdidas pode ocorrer devido à existência de um número reduzido de táxis disponíveis, tal como se evidencia ao domingo onde o *call center* tem uma perda de 13 em 140 chamadas recebidas em média, por análise do gráfico da figura 3.

Contudo, em regra, as chamadas são perdidas devido à incapacidade de resposta do *call center*, sendo a sua capacidade de resposta aumentada com recurso ao operador virtual implementado neste projeto.

Relativamente a uma comparação entre o número de chamadas perdidas antes e depois de colocar o IVR em funcionamento, torna-se difícil de realizá-la uma vez que depois da implementação do operador virtual este não foi utilizado nos dias críticos do *call center*, isto é, no Ano Novo e nas festas da cidade. Como tal, a comparação relativamente a este ponto da performance é difícil de realizar uma vez que é nestes dias que se vai evidenciar uma diferença notória.

CAPÍTULO 7: CONCLUSÃO

CAPÍTULO 7: CONCLUSÃO

Ao longo deste relatório é apresentada a estrutura e implementação do sistema IVR desenvolvido no âmbito do estágio curricular, sendo descrito o conjunto de ferramentas que auxiliaram a sua criação.

É também feita uma explicação de todos os requisitos que o IVR criado deve conter, dando-se ênfase às funcionalidades que este possui. De igual forma todas as estruturas utilizadas para o seu desenvolvimento são descritas para se perceber a sua utilidade no trabalho realizado.

Foi também feita uma análise dos resultados, para desta forma se descobrir quais as vantagens que este tipo de sistema traz para uma *call center*, que tal como se esperava são muitas, devido ao potencial que um operador virtual tem num *call center*.

As funcionalidades descritas na secção 3.1, foram todas concretizadas no decorrer do estágio, contudo, esta operadora virtual poderá vir a ter mais funções, ou seja, este projeto tem potencial para otimizações, visto que a operadora em causa pode vir a ter mais funcionalidades a si associadas.

O projeto desenvolvido teve como principal objetivo a criação de um sistema de IVR, ou seja, a criação de uma operadora virtual, que visa libertar os restantes operadores de certas funções como forma de aumentar o nível de eficiência dos *call center*, onde este sistema se encontra implementado.

APÊNDICE A

Acrónimos

IVR – Inteactive Voice Response

DTMF – Dual-Tone Multi-Frequency

PBX – Private Branch Exchange

CLI – Caller Line Identification

SPI - Session Initiation Protocol

REFERÊNCIAS

- [1] “Geolink Mobile Intelligence,” [Online]. Available: <http://www.geolink.pt/aboutUs.html>. [Acedido em abril 2014].
- [2] “TechTerms,” [Online]. Available: <http://www.techterms.com/definition/ivr>. [Acedido em abril 2014].
- [3] “Specialty Answering Service,” [Online]. Available: <http://www.specialtyansweringervice.net/library-books/what-is-the-voder/>. [Acedido em abril 2014].
- [4] “Princeton,” [Online]. Available: http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Dual-tone_multi-frequency.html. [Acedido em abril 2014].
- [5] “inContact,” [Online]. Available: <http://www.incontact.com/call-center-software/interactive-voice-response-ivr>. [Acedido em abril 2014].
- [6] “Enghouse Interactive,” [Online]. Available: http://www.syntellect.com/pages/products/communications_portal.aspx. [Acedido em abril 2014].
- [7] “Talkdesk,” [Online]. Available: <http://www.talkdesk.com/call-center-software/ivr-interactive-voice-response/>. [Acedido em abril 2014].
- [8] “About.com - Voice Over IP,” [Online]. Available: <http://voip.about.com/od/glossary/g/whatisPBX.htm>. [Acedido em Maio 2014].
- [9] “Asterisk,” [Online]. Available: <http://www.asterisk.org/get-started>. [Acedido em abril 2014].
- [10] “Voip-Info: Asterisk config queues.conf,” [Online]. Available: <http://www.voip-info.org/wiki/view/Asterisk+config+queues.conf>. [Acedido em abril 2014].
- [11] “Voip-Info: Asterisk config sip.conf,” [Online]. Available: <http://www.voip-info.org/wiki/view/Asterisk+config+sip.conf>. [Acedido em abril 2014].
- [12] “Voip-Info: Asterisk auto-dial out,” [Online]. Available: <http://www.voip-info.org/wiki/view/Asterisk+auto-dial+out>. [Acedido em abril 2014].
- [13] “Voip-Info: Asterisk config extensions.conf,” [Online]. Available: <http://www.voip-info.org/wiki/view/Asterisk+config+extensions.conf>. [Acedido em abril 2014].
- [14] “The Programming Language LUA,” [Online]. Available: <http://www.lua.org/>. [Acedido em abril 2014].

[15] "Voip-Info: Asterisk standard extensions," [Online]. Available: <http://www.voip-info.org/wiki/view/Asterisk+standard+extensions>. [Acedido em abril 2014].