

Enterprise Distributed System

TicketFactory



2º Projeto de TDIN 2019/2020

Regente: António Miguel Pontes Pimenta Monteiro

Mestrado Integrado em Engenharia Informática e Computação

João Pedro Viveiros Franco – up201604828

Tomás Nuno Fernandes Novo - up201604503

Índice

1 – Introdução	3
2 – Arquitetura	4
2.1 – <i>Website</i>	4
2.2 – <i>Service</i>	5
2.3 – <i>Server</i>	5
2.4 – <i>Solver</i>	5
2.5 – <i>Department</i>	5
2.6 – Base de Dados	6
3 – Funcionalidades e Testes	7
3.1 – Submissão de um <i>trouble ticket</i>	7
3.2 – Visualização de <i>trouble tickets</i> submetidos (<i>worker</i>)	7
3.3 – Autenticação como Solver/Department	7
3.4 – Visualização de <i>trouble tickets</i> submetidos (<i>solver</i>)	8
3.5 – <i>Self-assign</i> de <i>trouble tickets</i>	8
3.6 – Submissão de um <i>secondary ticket</i>	8
3.7 – Visualização de <i>secondary tickets</i> submetidos (<i>solver/department</i>)	9
3.8 – Resposta a <i>secondary tickets</i> submetidos	9
3.9 – Resolução de <i>trouble tickets</i>	9
4 – Modo de Funcionamento	11
5 – <i>Build, Install and Run</i>	21
6 – Conclusão	21

1 – Introdução

No âmbito da unidade curricular Tecnologias de Distribuição e Integração, foi-nos proposto como 2ª trabalho prático o desenvolvimento em **C#** de uma **Service Oriented Application**, nomeadamente de um **Enterprise Distributed System**. Para tal, usufruímos do **Visual Studio**, que nos permitiu não só a criação de uma interface gráfica (**GUI**), que facilita a interação dos utilizadores, como também o uso de um **Central REST service e web** em **.NET**

Resumindo um pouco a aplicação que desenvolvemos, cada **worker** submete um **trouble ticket** na **IT Web Application**, sendo que este *ticket* é analisado por um **solver** do **IT Department**, que tenta solucionar a questão submetida. No caso de existirem dúvidas aquando da resolução do problema, os *solvers* podem criar um **SecondaryTicket** (relativo ao *ticket* inicialmente submetido) que é enviado para um departamento secundário. o departamento secundário responde à pergunta submetida pelo *solver*, sendo possível a submissão de novas questões ao mesmo ou a outros departamentos. Após receber as respostas dos outros departamentos, ou caso o *solver* não possua dúvidas acerca do *ticket* inicialmente submetido, este resolve-o enviando um mail para o *worker* que o submeteu.

2 – Arquitetura

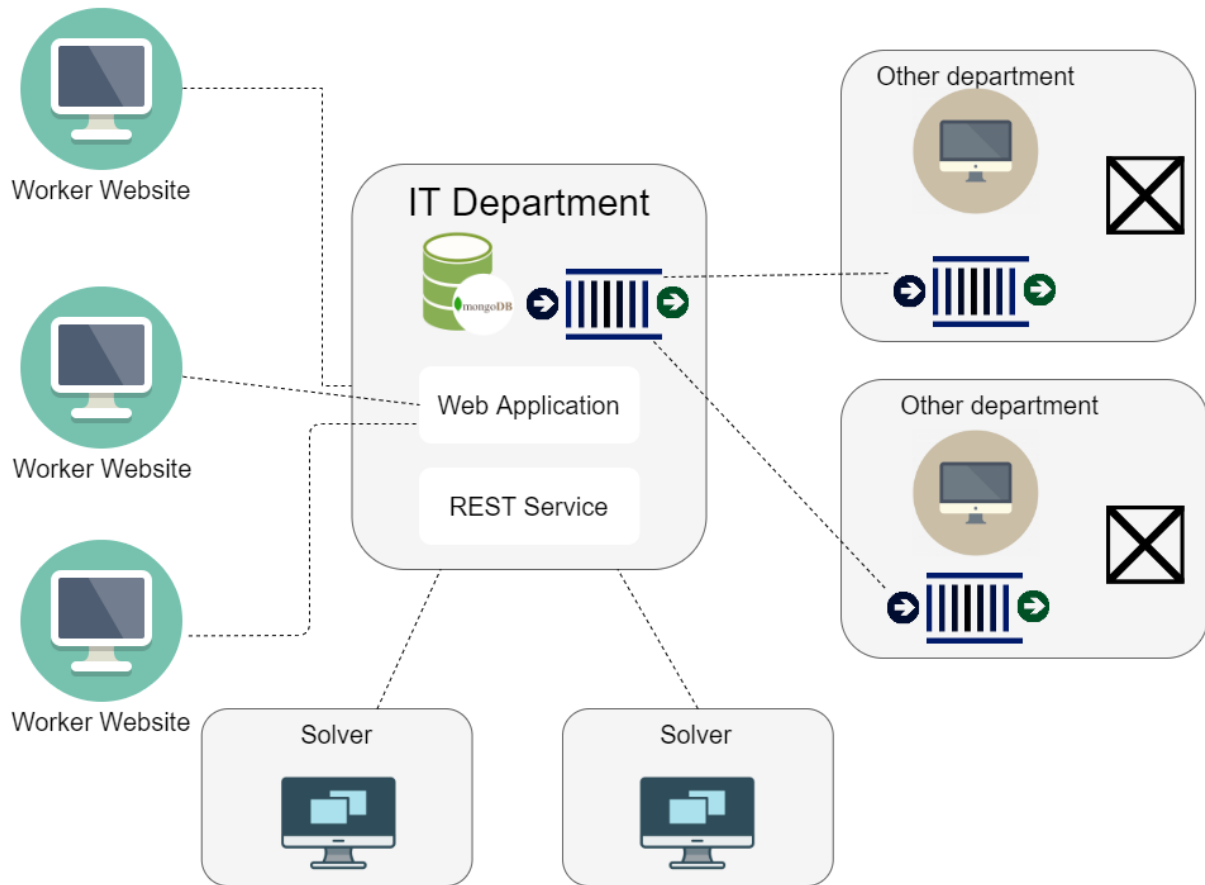


Fig.1 – Arquitetura da aplicação

A solução que concebemos encontra-se dividida em diversos módulos: **Solver**, **Department**, **Service**, **Server** e **Website**.

2.1 – Website

O módulo **Website** consiste num **ASP.NET web forms application** que usufrui do módulo **Service**, invocando métodos deste para realizar a submissão dos *trouble tickets* e para permitir a visualização de todos os tickets submetidos até à data por um *worker*.

A interface deste módulo é simplista e foi criada com o fim de permitir uma boa interação por parte dos *workers*.

2.2 – Service

Este módulo consiste num **REST WCFService** que é usado pela *Web Application*. Com o fim de permitir que os restantes módulos invoquem métodos deste serviço, é utilizada uma **proxy** que se encontra no módulo **Server**.

O módulo em questão possui uma base de dados explicitada no módulo **2.6**, na qual são armazenados os *trouble tickets*.

2.3 – Server

Por sua vez, o módulo do servidor consiste numa simples .NET Console Application que usa a classe `WebServiceHost` como um REST Web server com o fim de dar *host* ao serviço.

Neste componente, encontra-se a proxy responsável por permitir que os diferentes módulos acessem e usem o módulo do serviço.

É necessário correr este módulo como administrador devido a limitações de segurança do *Windows*.

2.4 – Solver

Este módulo consiste numa **Windows Forms App** na qual o utilizador se identifica e realiza diversas ações, como visualizar os *trouble tickets* submetidos, dar *self-assign* num *trouble ticket*, criar um *ticket* secundário e enviá-lo para outro departamento e resolver um *ticket* através do envio de um email para o submissor do *ticket* inicial.

O *solver* acede à base de dados do servidor para poder visualizar quer os *trouble tickets* quer os secundários submetidos, bem como alterar o estado dos *trouble tickets* conforme as ações que realiza.

Quando comunica com outros departamentos, a comunicação é feita através de uma *queue* assíncrona.

2.5 – Department

À semelhança do Solver, este módulo consiste numa **Windows Forms App** na qual o utilizador se autentica e realiza diversas ações à volta dos *tickets* secundários submetidos pelos *solvers*: o *user* pode visualizar aqueles que foram enviados para o departamento ao qual o utilizador pertence, bem como responder à questão neles contida.

Cada instância de departamento recebe os *tickets* secundários através de uma

queue assíncrona, inserindo posteriormente na base de dados relativa a cada departamento as informações relativos ao ticket recebido.

2.6 – Base de Dados

Dada a necessidade de guardar informação relativa a cada ticket, foi criada e integrada no projeto uma **Base de Dados MongoDB**.

O servidor possui uma base de dados à qual acedem os *solvers*, enquanto que cada departamento tem acesso a uma pequena base de dados local para os *secondary tickets*.

O código desenvolvido relativo à DB encontra-se no ficheiro **Database.cs**.

Para correr a *database*, foi criada a *script openDB.bat*, sendo necessário corrê-la antes de iniciar o servidor.

3 – Funcionalidades e Testes

3.1 – Submissão de um *trouble ticket*

Os *workers* usufruem desta funcionalidade no Website desenvolvido, sendo que é necessária a identificação do mesmo (**nome** e **email**) aquando da submissão. Além desta “autenticação” é necessária a atribuição de um título e de uma descrição para o *ticket* em questão.

Com o fim de testar esta funcionalidade, foram submetidos diversos *tickets* com diversos utilizadores, sendo possível verificar que a introdução na base de dados é realizada com sucesso e que cada *ticket* possuiu um identificador único.

3.2 – Visualização de *trouble tickets* submetidos (*worker*)

Cada *worker* tem a possibilidade de visualizar todas as suas submissões. Para tal, apenas é necessário o preenchimento do campo de nome no Website, tendo como finalidade a identificação.

Para testar a referida funcionalidade, foram efetivadas diversas submissões com diversos *workers*, sendo verificado que esta visualização ocorria sem qualquer obstáculo.

3.3 – Autenticação como *Solver/Department*

Quando uma GUI de um *solver* é iniciada, é requerida a identificação do *solver*, nomeadamente o seu **nome**. Este identificador é associado a um *ticket* quando o solucionador decide tentar resolver o mesmo.

Por sua vez, aquando da inicialização GUI de um membro de outro departamento, a identificação consiste no seu **nome** e **identificador** do departamento. O identificador do **departamento** é necessário na submissão dos *tickets* secundários, como explicado no ponto 3.6.

Objetivando testar esta funcionalidade, foram gerados diversos *tickets* e *secondary tickets* e inicializadas diversas instâncias dos dois tipos de interfaces. Averiguou-se que quando um *solver* decide ficar responsável pela resolução de um *ticket*, o seu nome é associado com sucesso, assim como apenas os membros do departamento para o qual os *secondary tickets* foram encaminhados podem ver os mesmos.

3.4 – Visualização de *trouble tickets* submetidos (solver)

Cada *solver* possui a possibilidade de visualizar na sua interface todas as submissões, sendo ainda possível filtrar esta visualização: podem ser observados os *tickets* que cada *worker* submeteu, os *tickets* que ainda não têm *solver* atribuído ou todos os *tickets*.

Com o fim de testar esta referida funcionalidade, foram gerados vários *tickets* por diversos *workers*, sendo que alguns foram atribuídos e outros não. Concluiu-se que a visualização com qualquer dos filtros funciona como previsto.

3.5 – *Self-assign* de *trouble tickets*

Cada *solver*, enquanto visualiza as submissões dos *workers*, tem a possibilidade de escolher que *tickets* quer solucionar dando *self-assign* individualmente a cada *trouble ticket*, sendo que estes não só aparecem numa área restrita a cada “solver” como também passam do estado **Unassigned** para **Assigned**, sendo atualizada na base de dados o referido estado e associado o nome do solucionador. A partir do momento em que é atribuído o *ticket* ao solucionador, este tem a responsabilidade de o resolver.

Com o propósito de testar a funcionalidade em questão, foram gerados vários *tickets* por diversos *workers* que posteriormente sofreram *self-assign* de diferentes instâncias de classes *solvers*. Concluiu-se que a atribuição é feita com sucesso e os *tickets* ficam disponíveis na área “**My Tickets**” pertencente a cada *solver*.

3.6 – Submissão de um *secondary ticket*

Na sua interface, na secção “**Ask Question**”, no caso de existirem dúvidas na resolução de um *trouble ticket*, os *solvers* podem submeter um **ticket secundário** que é enviado para outros departamentos com o fim de obter auxílio na resolução, sendo ainda possível escolher qual o departamento que vai responder à questão.

Intencionando verificar esta funcionalidade, foram submetidos diversos *secondary tickets* por parte de diversos *solvers*. Foi possível verificar que a introdução na base de dados é feita com sucesso e que apenas os departamentos preferidos por cada *solver* em cada *ticket* secundário recebem os ditos cujos.

3.7 – Visualização de *secondary tickets* submetidos (solver/department)

Cada *solver* possui a possibilidade de visualizar na sua interface todas as suas submissões de *tickets* secundários. Por sua vez, na sua interface, os membros do departamento podem observar todos os *secondary tickets* enviados para o departamento a que pertencem.

Com a meta de verificar esta funcionalidade, foram submetidos diversos *secondary tickets* por parte de diversos *solvers*. Foi possível verificar que a introdução na base de dados é feita com sucesso e que a visualização ocorre sem obstáculos. Apenas os departamentos preferidos por cada *solver* em cada *ticket* secundário recebem e podem ver os ditos cujos.

3.8 – Resposta a *secondary tickets* submetidos

Os membros de um departamento podem seleccionar um *ticket* secundário submetido (através do seu ID) e estabelecer uma espécie de conversação com o *solver* que submeteu o *ticket* em questão. O *secondary ticket* tem um ID único, mas tem também associado um ID de um *ticket* e uma questão, à qual o membro do departamento responde e reencaminha para o *solver*. No caso de o *solver* não ficar esclarecido, pode submeter um novo *ticket* secundário associando o identificador do *trouble ticket* inicial. Os membros do departamento, no caso de haver uma submissão nova para o mesmo *ticket*, conseguem visualizar as perguntas e respostas anteriores visto que a secção onde é inserida a resposta ser uma espécie de chat.

Quando uma resposta é enviada por um membro de outro departamento, o *solver* que submeteu o *ticket* secundário recebe uma notificação que resulta de um constante **polling** que ocorre na interface deste.

Tendo como alvo verificar o funcionamento desta funcionalidade, foram submetidos diversos *secondary tickets* por parte de diversos *solvers*. Foi possível verificar que a resposta é efectuada com sucesso quando há novas submissões relativas ao mesmo *trouble ticket* ou quando as submissões são independentes entre si, sendo que o *solver* é notificado quando ocorre uma resposta a dos *tickets* secundários submetidos.

3.9 – Resolução de *trouble tickets*

Na secção “**Resolve Ticket**”, os *solvers* podem resolver um *trouble ticket* enviando um *email* para o *worker* que o submeteu.

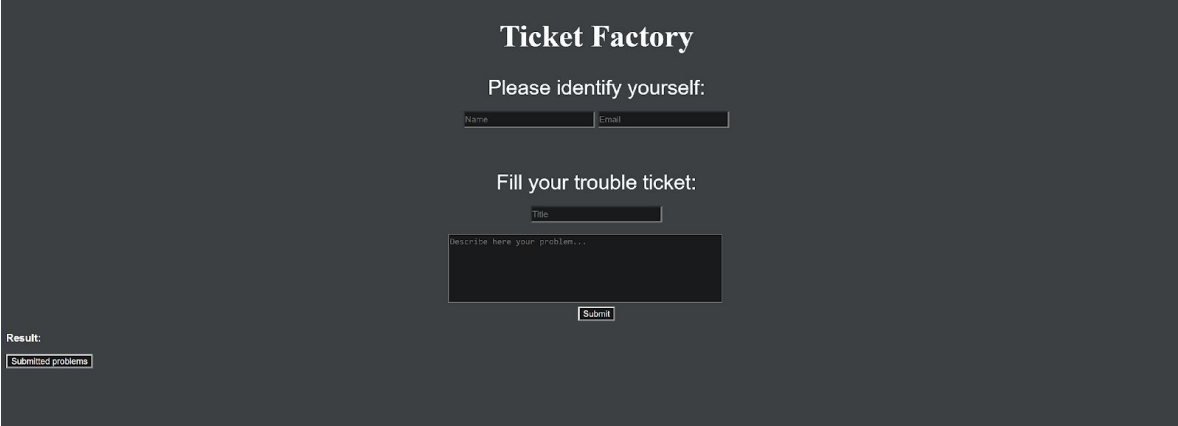
Tendo como fim verificar o funcionamento desta funcionalidade, foram enviados

diversos emails para os diversos *solvers*, verificando-se que o estado dos *tickets* se altera, como esperado, para **Solved**, e que a receção dos emails é efectuada com êxito.

4 – Modo de Funcionamento

O modo de funcionamento varia conforme os diversos utilizadores da aplicação que concebemos.

Princiando pelo **Website**, nele é possível submeter um *trouble ticket*. Como forma de identificar quem submete, e para ser possível posteriormente mandar email com a solução, é necessária a submissão de um nome e de um email por parte dos *workers*. Após inserir um título e uma descrição, carregando no botão “**Submit**”, o *ticket* é enviado para a base de dados do **IT Department**.



The screenshot shows a web form titled "Ticket Factory". It has two main sections: "Please identify yourself:" with input fields for "Name" and "Email", and "Fill your trouble ticket:" with a "Title" input field and a larger "Describe here your problem..." text area. A "Submit" button is located below the text area. At the bottom left, there is a "Result:" section with a "Submitted problems" button.

Fig. 2 – Website: submissão de um trouble ticket

Ainda no **Website**, é possível verificar as submissões efectivadas até ao momento preenchendo o campo “**Name**”.



This screenshot shows the same "Ticket Factory" form as in Fig. 2, but with the "Submitted problems" button clicked. Below the form, a table displays the list of submitted tickets. The table has columns for id, username, date, title, description, status, and solver.

id	username	date	title	description	status	solver
5ecbf0bd38baaf123a00d1fb6	Tomás Novo	23/05/2020 16:22:21	Email problem	Hello, I'm a solver and lately I've been receiving the ticket solutions on a mail that I don't use anymore ! Can you help me pls ?	Unassigned	

Fig. 3 – Website: visualização dos trouble tickets submetidos por um worker

Por sua vez, quando o **Solver** inicia a sua interface, a sua autenticação consiste na inserção do seu nome.

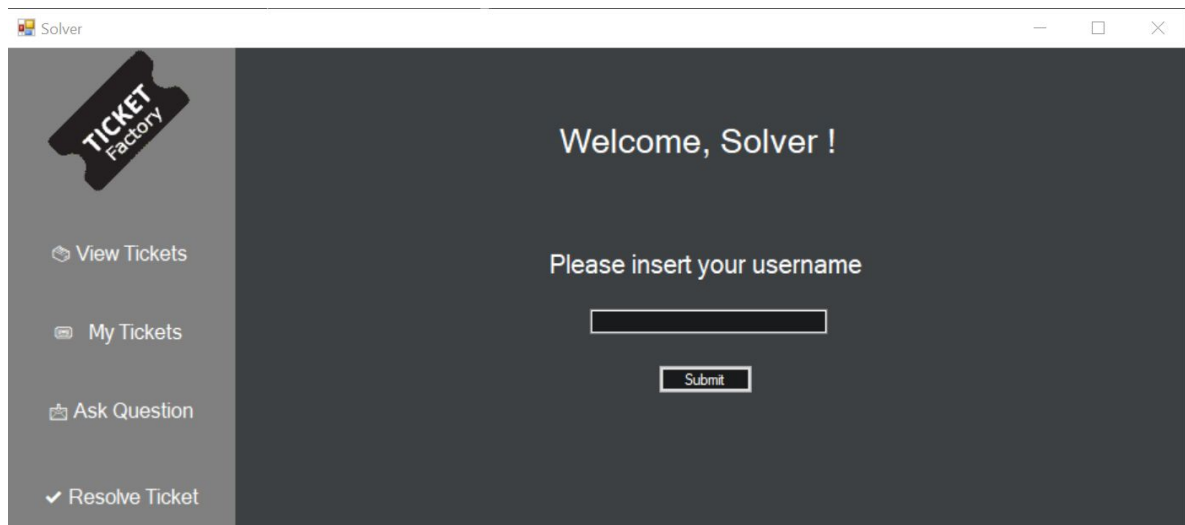


Fig. 4 –Solver: Ecrã de Autenticação

Após se identificar, o *sol/ver* está pronto a usufruir da **navbar** que tem ao seu dispor na interface.

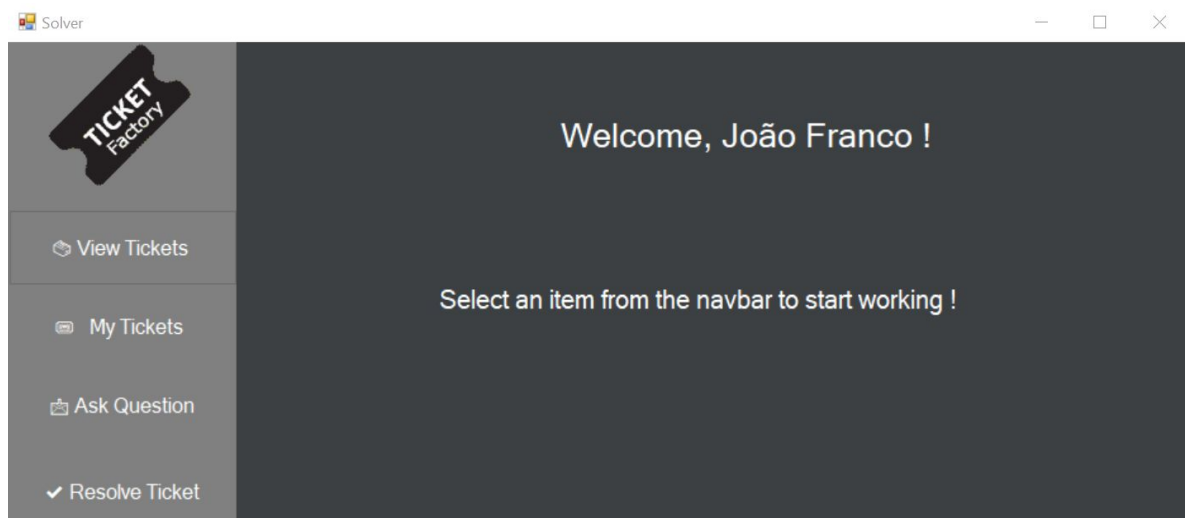


Fig. 5 –Solver: Ecrã pós-Autenticação

Se o solucionador desejar ver os *tickets*, pode definir uma preferência de visualização, podendo observá-los todos, ou por username ou pelo estado **Unassigned**.

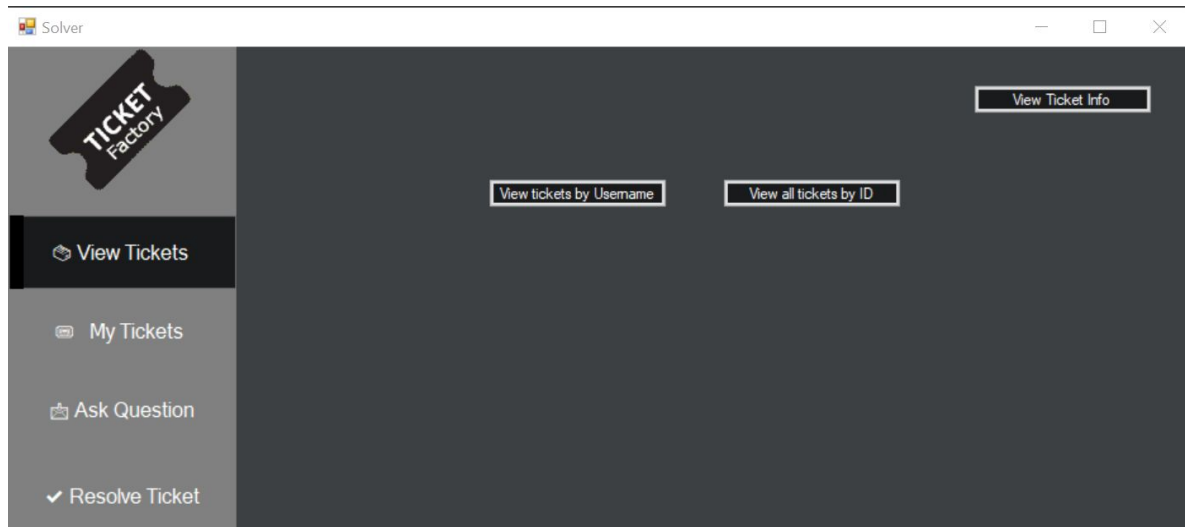


Fig. 6 –Solver: Ecrã View Tickets

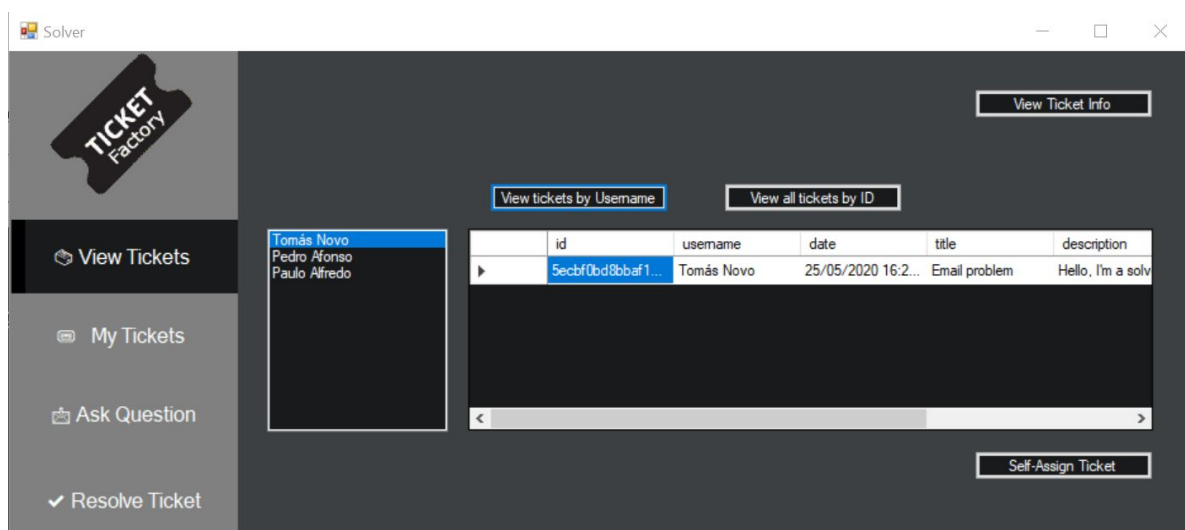


Fig. 7 –Solver: Ecrã View Tickets by Username

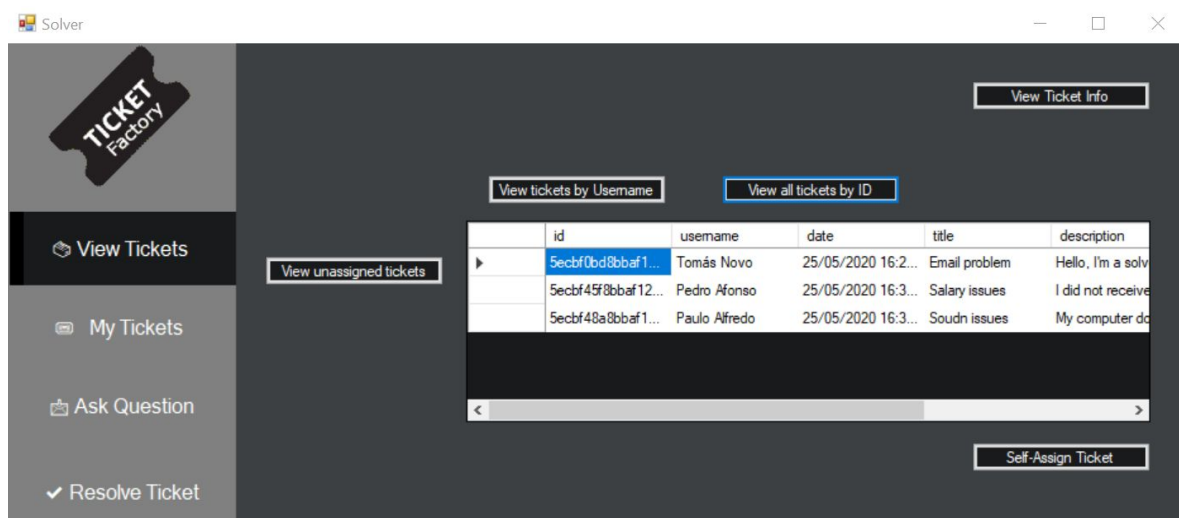


Fig. 8 –Solver: Ecrã View all Tickets by ID

Esta visualização tem como fim permitir uma observação geral dos *tickets*. Caso o *solver* pretenda uma visão mais específica sobre um *ticket*, pode seleccionar o seu *id* na tabela e carregar no botão “**View Ticket Info**”.

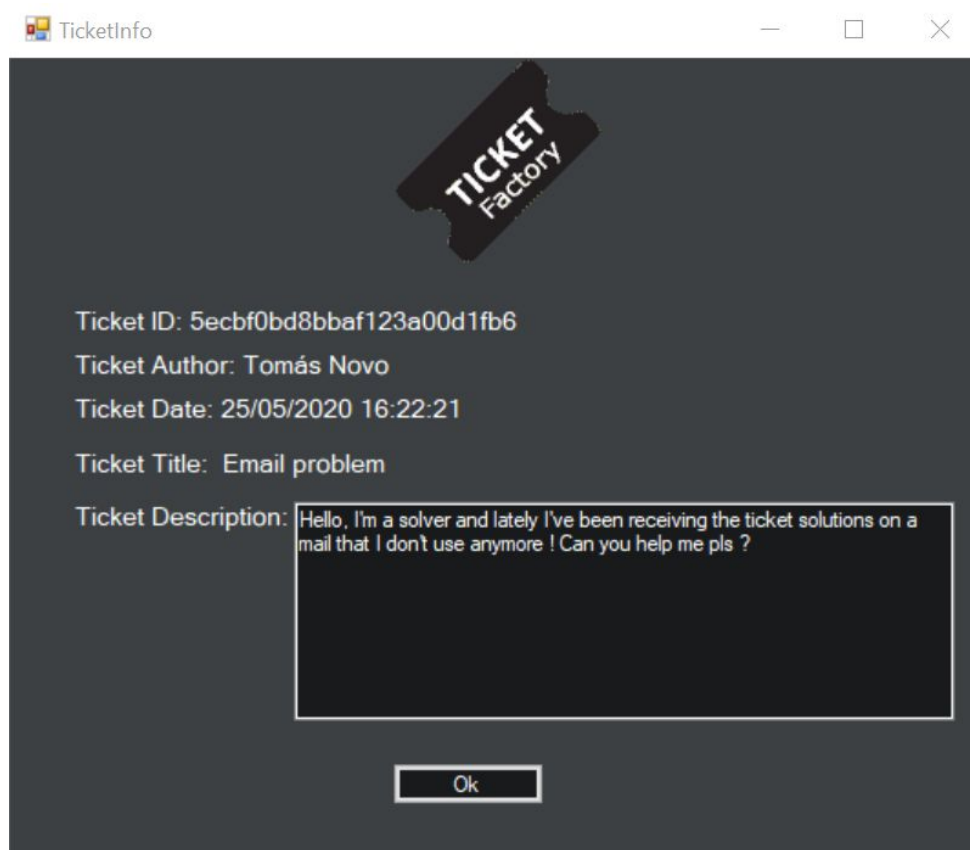


Fig. 9 –Solver: Ecrã View Ticket Info

O *solver* pode decidir dar *self-assign* num *ticket* carregando no botão que possui esse fim. Quando isto ocorre, o estado do *trouble ticket* passa para **Assigned** e o campo *solver* é preenchido com o nome deste. A partir do momento em que é atribuído, o *ticket* pode ser visto por outros *solvers* mas não poderá ser atribuído a nenhum outro *solver*. Os *tickets* que já estão atribuídos ao *solver*, aparecem na secção “**My Tickets**” da barra de navegação.

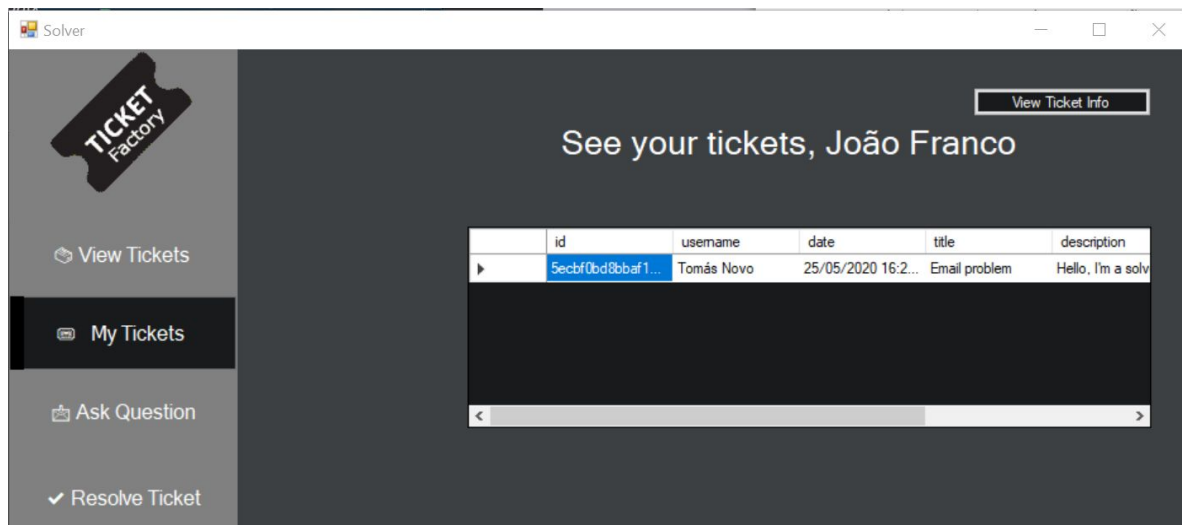


Fig. 10 –Solver: Ecrã My Tickets

Seleccionando o *id* de um *ticket* nesta secção, o *solver* pode criar uma questão relativa ao mesmo que será enviada para outro departamento ou resolvê-lo.

Na secção “**Ask Question**” o *solver* pode preencher o campo “**Department ID**” para informar que departamento receberá o *ticket* secundário. Preenchendo o campo de título e da questão em si, o *secondary ticket* é enviado para o departamento desejado.

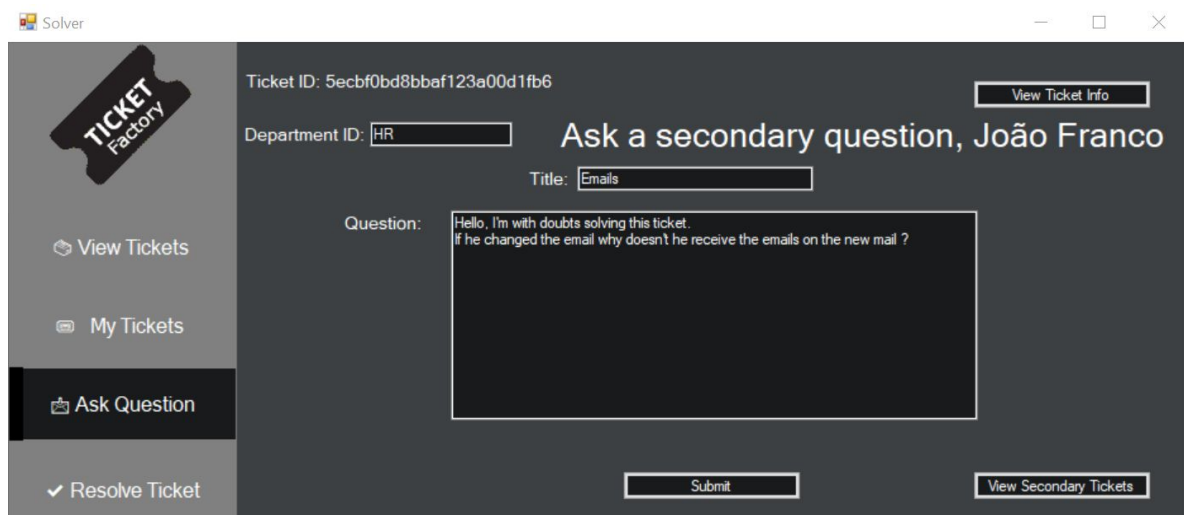


Fig. 11 –Solver: Ecrã Ask Question

Carregando no botão “**View Secondary Tickets**”, o *solver* pode ver todos os *tickets* secundários submetidos. É possível também preencher o campo **Ticket ID** e visualizar todos os *secondary tickets* relativos ao mesmo *trouble ticket*.

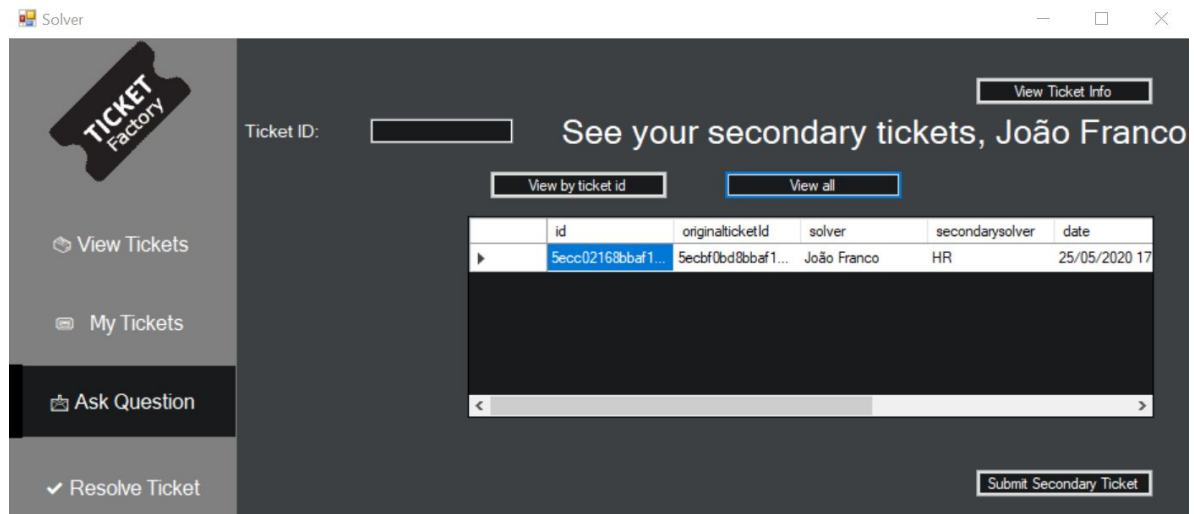


Fig. 12 – Solver: Ecrã visualização secondary tickets submetidos

As interfaces das figuras 13, 14, 15 e 16 são relativas aos membros de um departamento.

Como autenticação, é requerido o nome do membro e nome do departamento.

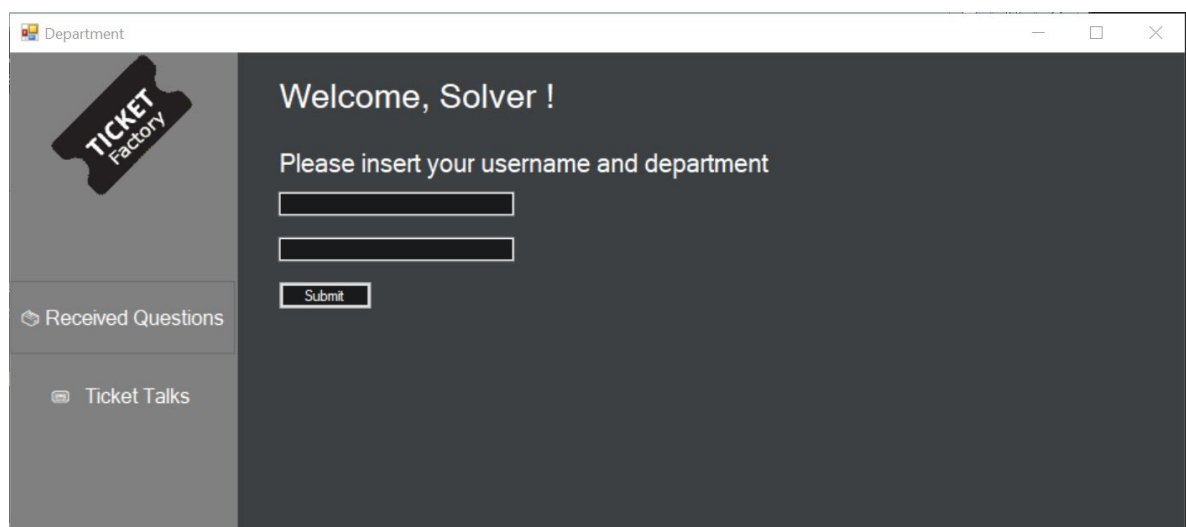


Fig. 13 – Department: Ecrã Autenticação

Após se identificar, o membro está pronto a usufruir da **navbar** que tem ao seu dispor na interface.

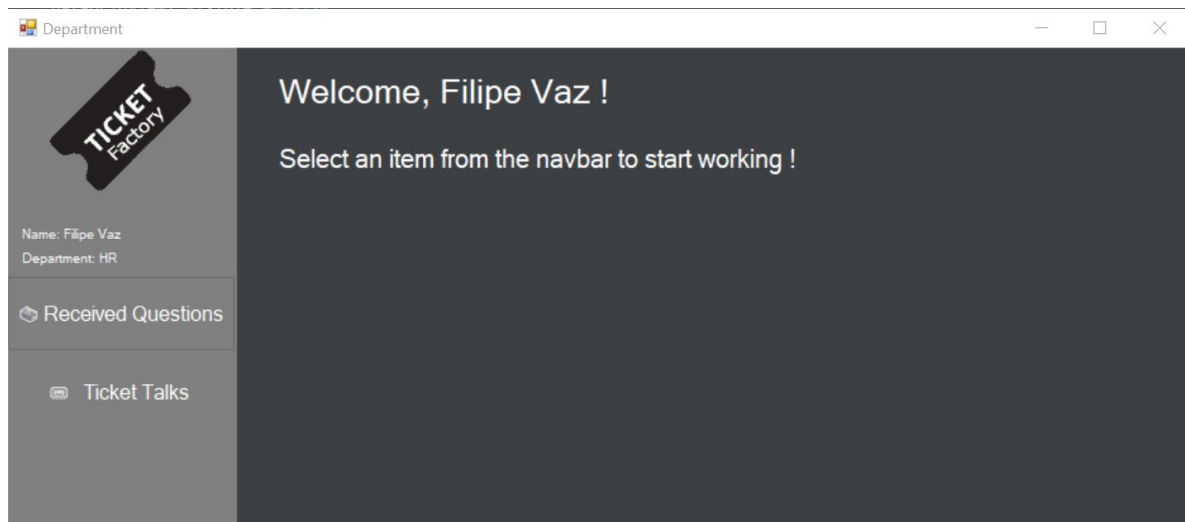


Fig. 14 – Department: Ecrã pós-Authenticação

Na secção “**Received Questions**”, o membro do departamento consegue ver todos os secondary tickets que o respetivo departamento recebeu, podendo ainda observar as informações relativas ao *trouble ticket* inicial pressionando o botão “**View Ticket Info**”

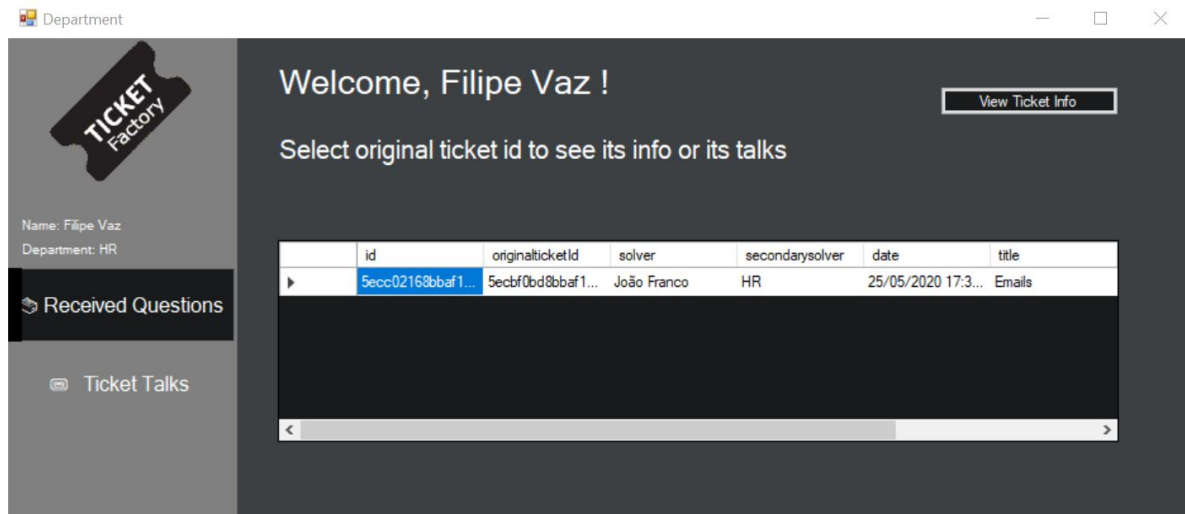


Fig. 15 – Department: Ecrã visualização secondary tickets

Selecionando o *id* do *ticket* original e carregando na secção da *navbar* “***Ticket Talks***”, o membro consegue ver todas as questões e respostas relativas a esse *ticket*, conseguindo submeter uma nova resposta no caso de haver uma pergunta por responder.

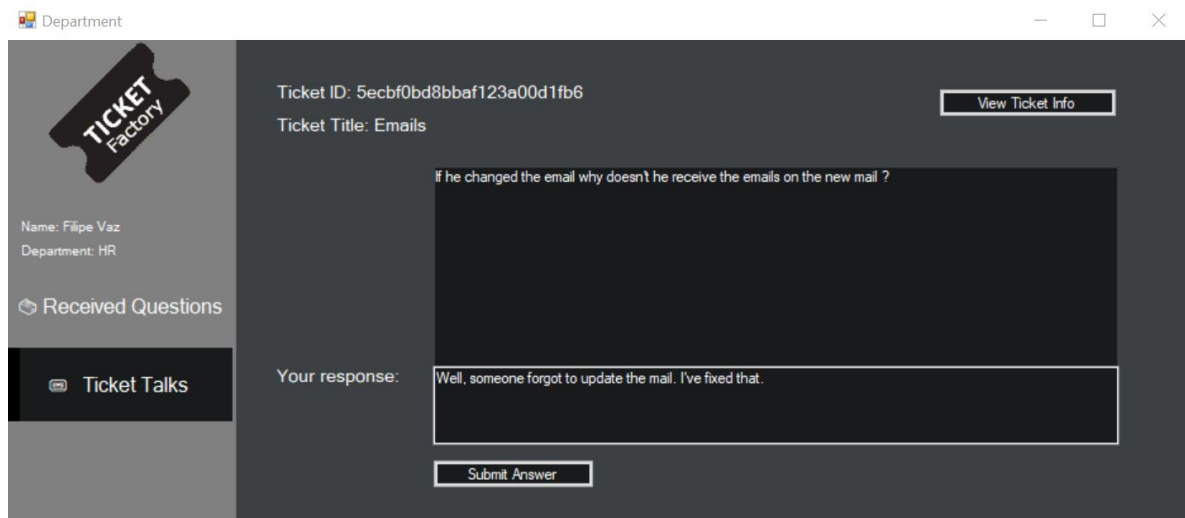


Fig. 16 – Department: Ecrã pergunta-respostas de secondary tickets

Após a resposta ser submetida, o membro do departamento é notificado do sucesso do envio da mesma, assim como o solver que submeteu o *SecondaryTicket*

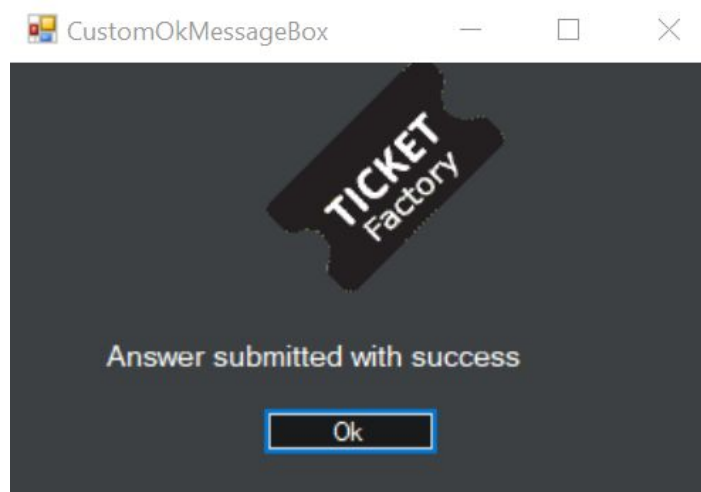


Fig. 17 – Department: Ecrã sucesso de envio de resposta secondary tickets

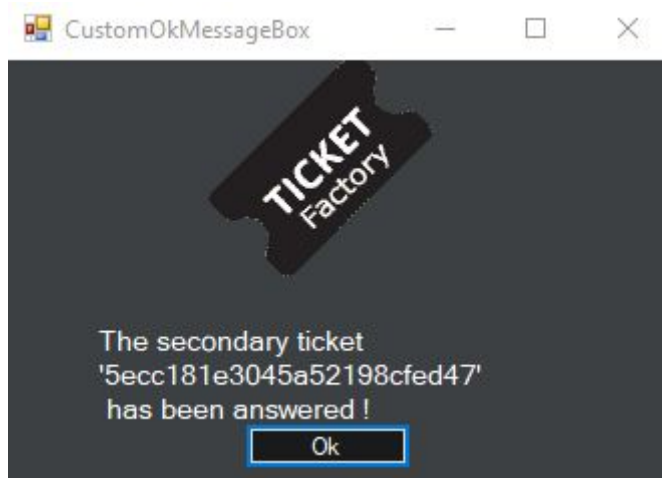


Fig. 18 – Solver: Ecrã sucesso de envio de resposta aos secondary tickets

Após a receção da resposta de um membro de outro departamento ou no caso de não ser necessário auxílio, um solver pode resolver o *ticket* em questão na secção “**Resolve Ticket**” da *navbar*. Após alterar o conteúdo do *email*, o email é enviado para o recetor que submeteu o *trouble ticket* e o estado deste passa para **Solved**,

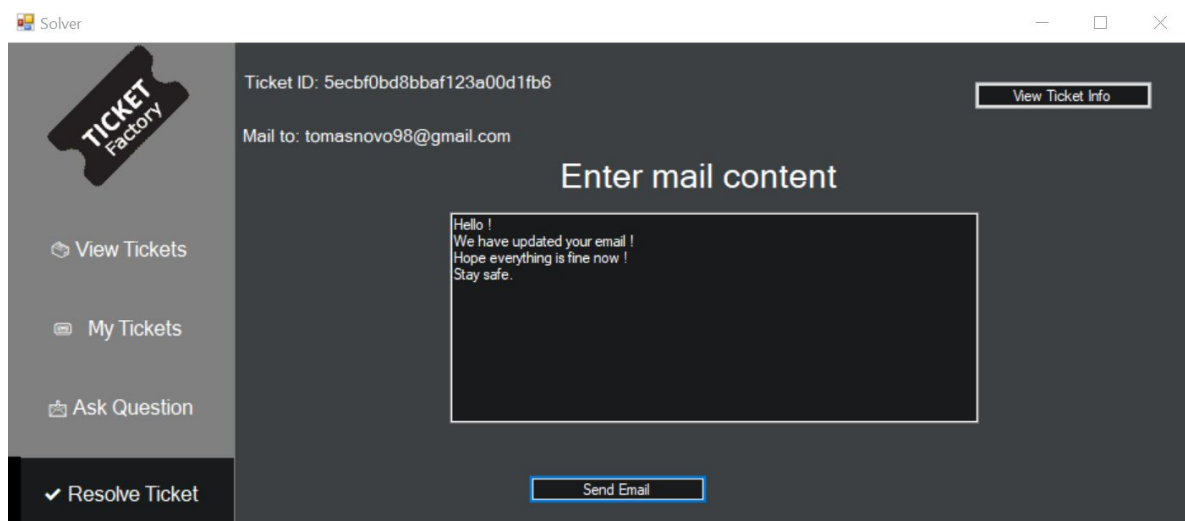


Fig. 19 – Department: Ecrã de resolução de um trouble tickets

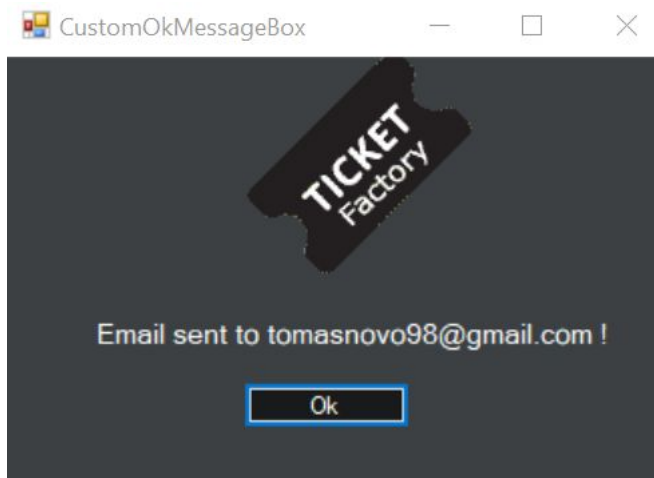


Fig. 20 – Department: Ecrã de sucesso na resolução de um trouble ticket

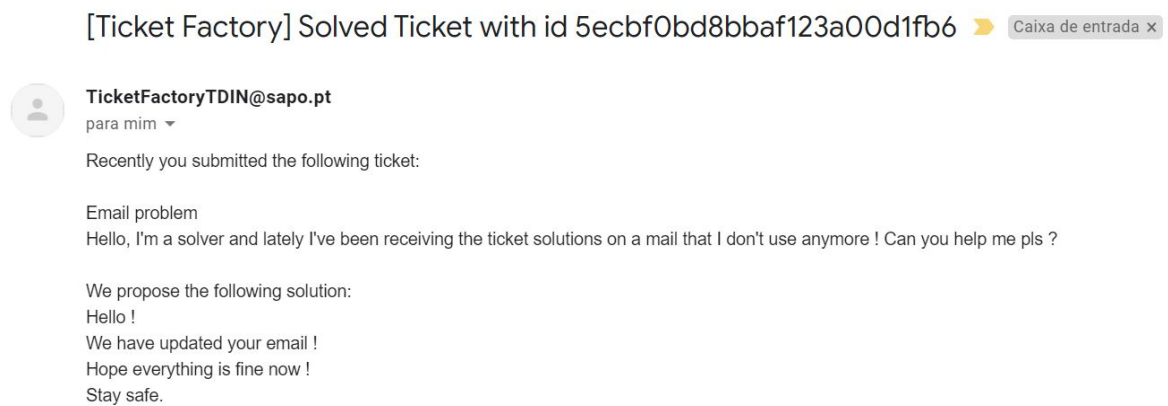


Fig. 21 – Worker: Receção de um email relacionado com um trouble ticket

5 – Build, Install and Run

Para usufruir da nossa aplicação, o primeiro passo é correr a *script openDB.bat* que se encontra no repositório. Posteriormente, é necessário abrir o **Visual Studio** como **administrador** e dar *build* ao projeto. Depois, corre-se o servidor, com o fim de dar host ao **REST WCFService**. Finalmente, abre-se uma instância **Solver** em modo **Debug** para o **Website** abrir na ligação <http://localhost:55165/>. Após ser encerrada, podem ser instanciadas diversos **Solvers** e diversos **Departments**.

6 – Conclusão

Em suma, a realização deste projeto foi bastante positiva para os elementos do grupo devido à aquisição de conhecimentos relativamente ao funcionamento de uma **Service Oriented Application** e à manipulação da linguagem **C#**.

A aplicação que desenvolvemos não só permite que diversos *workers* submitam os seus *trouble tickets* como também permite que diversos *solvers* os solucionem. Aquando da resolução, estes podem requisitar auxílio a outros departamentos através da submissão de um *secondary ticket*.

Cada *ticket* e *ticket* secundário são arquivados na base de dados, sendo que os primeiros ficam alojados na base de dados do IT Department enquanto que os segundos são guardados e manipulados por outros departamentos.

Acreditamos que o produto final vai de encontro ao objetivado sendo que desenvolvemos com sucesso o sistema objetivado com uma interface intuitiva, permitindo aos diversos utilizadores realizarem as suas tarefas sem qualquer dificuldade.