

UNIVERSIDADE PAULISTA

Diego Felipe Martins Mendes RA N1268J0

Jayme Rossini Júnior RA D3840A4

UniSystem

Sistema de help desk.

SOROCABA – SP

2018

Diego Felipe Martins Mendes RA N1268J0

Jayme Rossini Júnior RA D3840A4

UniSystem

Sistema help desk.

Trabalho de conclusão de semestre
de curso de Analise de
desenvolvimento de sistema
apresentado à Universidade Paulista

Orientadores: Todos os professores
do semestre.

SOROCABA – SP

2018

CIP - Catalogação na Publicação

UniSystem Help desk / Diego Mendes...[et al.]. - 2019.
70 f. : il. color

Trabalho de Conclusão de Curso (Graduação) apresentado ao Instituto de Ciência Exatas e Tecnologia da Universidade Paulista, Sorocaba, 2019.

Área de Concentração: Sorocaba.

Orientador: Prof. Todos Professores.

1. UniSystem: Sistema de Help desk. I. Mendes, Diego. II. Professores, Todos (orientador).

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da Universidade Paulista
com os dados fornecidos pelo(a) autor(a).

Diego Felipe Martins Mendes RA N1268J0

Jayme Rossini Júnior RA D3840A4

UniSystem

Sistema de help desk.

Trabalho de conclusão de semestre
de curso de Análise de
desenvolvimento de sistema
apresentado à Universidade Paulista

Orientadores: Todos os professores
do semestre.

Aprovado em:

BANCA EXAMINADORA

_____/____/____

Prof: Wandivaldi Antonio Colla
Universidade Paulista

_____/____/____

Prof: Marcelo Tirolli
Universidade Paulista

_____/____/____

Prof: Randal Gasparini
Universidade Paulista

_____/____/____

Prof: Reverdan Almeida Springer
Universidade Paulista

RESUMO

O propósito da UniSystem é desenvolver sistema de “help desk”, que auxilia no gerenciamento de chamados quando abertos pelos usuários (funcionários) ao departamento de TI, dessa forma, os mantendo organizados e agilizando nos processos de manutenção; cada caso é atendido de acordo com o problema do usuário. O sistema foi desenvolvido utilizando linguagem de programação (plataforma .NET) e o banco de dados SQL Server

Palavra-Chave: Sistema. Usuário. UniSystem.

ABSTRACT

The purpose of UniSystem is to develop a "help desk", system that assists in the management of calls when opened by users (employees) to the IT (information technology) department, thereby keeping them organized and streamlining in the maintenance processes; each case is handled according to the user's problem. The system was developed using the programming language (.NET platform) and the SQL Server database.

Keyword: System. User. UniSystem.

SUMÁRIO

1.0 INTRODUÇÃO	8
2.0 UNISYSTEM.....	9
3.0 HELP DESK	10
3.1 Levantamento de Requisitos.....	10
3.2 Help desk Web	10
3.3 Sistema help desk desktop.....	16
4.0 FERRAMENTAS UTILIZADAS PARA O DESENVOLVIMENTO	17
4.1 Linguagens e Plataformas Utilizadas.....	17
4.2 SQL.....	17
4.2 ASP .NET	17
4.3 Windows Presentation Foundation (WFP).....	18
5.0 MODELO DE PROCESSO DE ENGENHARIA DE SOFTWARE	19
5.1 Objetivos da prototipação evolucionária e descartável.....	19
5.2 Pontos negativos da prototipação	19
5.3 Pontos positivos da prototipação	19
7.0 DIAGRAMA DE CASO DE USO	20
8.0 DIAGRAMA DE CLASSE	21
9.0 DIAGRAMA DE SEQUÊNCIA	22
9.1 Elementos Do Diagrama.....	22
9.2 Objetos	22
10.0 BANCO DE DADOS	24
10.1 Modelo conceitual.....	25
10.2 Modelo lógico.....	26
10.3 Modelo físico.....	27
11.0 CONCLUSÃO	28
12.0 REFERÊNCIAS.....	29
APÊNDICE	30
APÊNDICE A – Código ASP .NET.....	31
APÊNDICE B – Código WPF .NET	68

1.0 INTRODUÇÃO

UniSystem é uma empresa que desenvolve sistemas de help desk, para a melhoria do gerenciamento, controle, organização assim melhorando a operação do seu negócio.

O projeto a seguir de exemplo foi desenvolvido para um supermercado com intuito de facilitar os processos de suporte do departamento de TI.

O objetivo principal é gerenciar a abertura dos chamados realizado pelos usuários, melhorando no tratamento de suporte pelo departamento de TI.

O sistema foi desenvolvido utilizando linguagem de programação (plataforma .NET) e os dados estão sendo armazenados no SQL Server, seguindo um padrão de sistema CRUD.

2.0 UNISYSTEM

O propósito da UniSystem é desenvolver sistemas de help desk, para a melhoria do gerenciamento, controle, organização e agilidade nos processos de suporte das demandas, aprimorando a operação do seu negócio. Os sistemas são desenvolvidos utilizando linguagem de programação (plataforma .NET) e os dados são armazenados no SQL Server.

3.0 HELP DESK

Help desk se refere a um serviço de atendimento aos clientes que procuram esclarecimentos sobre dúvidas, soluções para problemas técnicos relacionados a telefonia, informática, tecnologia da informação.

O serviço de help desk centraliza o recebimento das solicitações dos clientes, o que facilita as respostas e resolução dos problemas. Este serviço pode ser realizado tanto dentro de uma empresa, quanto externamente com a prestação de serviços aos usuários através de um sistema de call center ou gerenciamento de ocorrências.

3.1 Levantamento de Requisitos

Um supermercado entrou em contato com a UniSystem, solicitando um sistema de help desk para facilitar os processos de suporte do departamento de TI.

O supermercado possui as seguintes ferramentas:

- Hardware (Desktop, Notebook, Servidor, PDV)
- Rede (Switch, Modem/Roteador)
- Telefonia (PABX)
- Impressoras
- Softwares (Office, Sistemas interno, Sistema dos PDVs)

A principal proposta é facilitar a abertura dos chamados pelos usuários de acordo com as ferramentas utilizadas por eles, e a melhoria no tratamento pelo departamento de TI, realizando feedbacks, alterações de status do chamado de acordo com as ações tomadas referente ao problema.

3.2 Help desk Web

As funções do sistema baseiam-se em cadastramento de usuários, setores, analistas, chamados e atendimento dos chamados.

Os administradores têm permissão para:

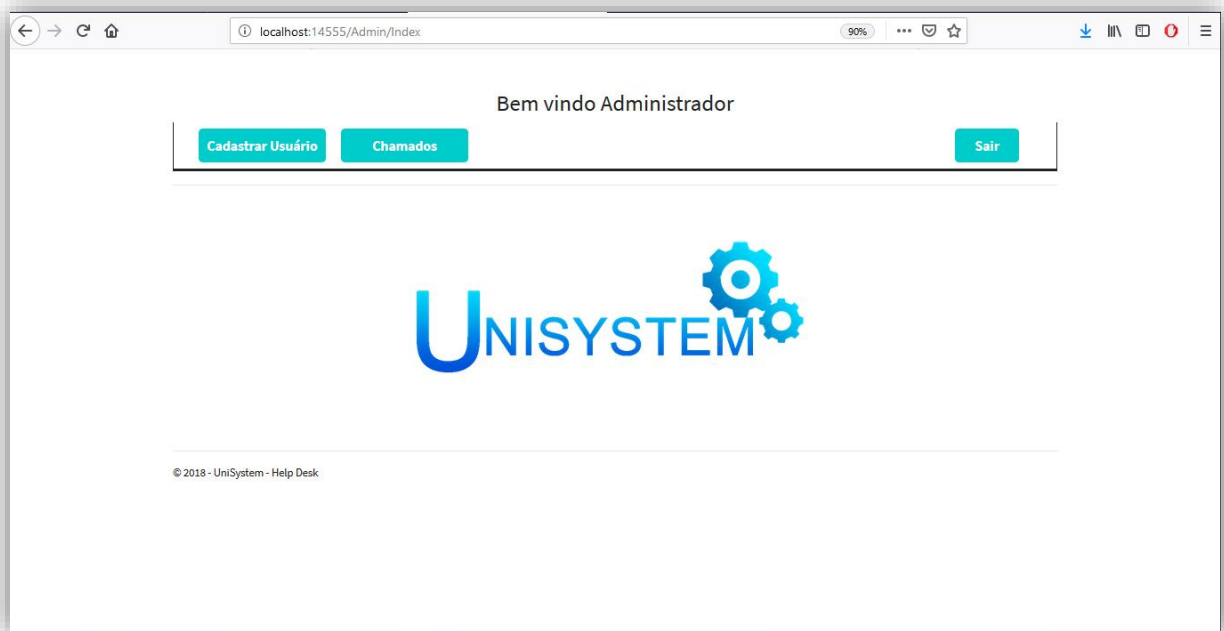
- Cadastro de usuários, setores e analistas
- Atendimento do chamado

Os Usuários têm permissão para:

- Cadastro de chamados.

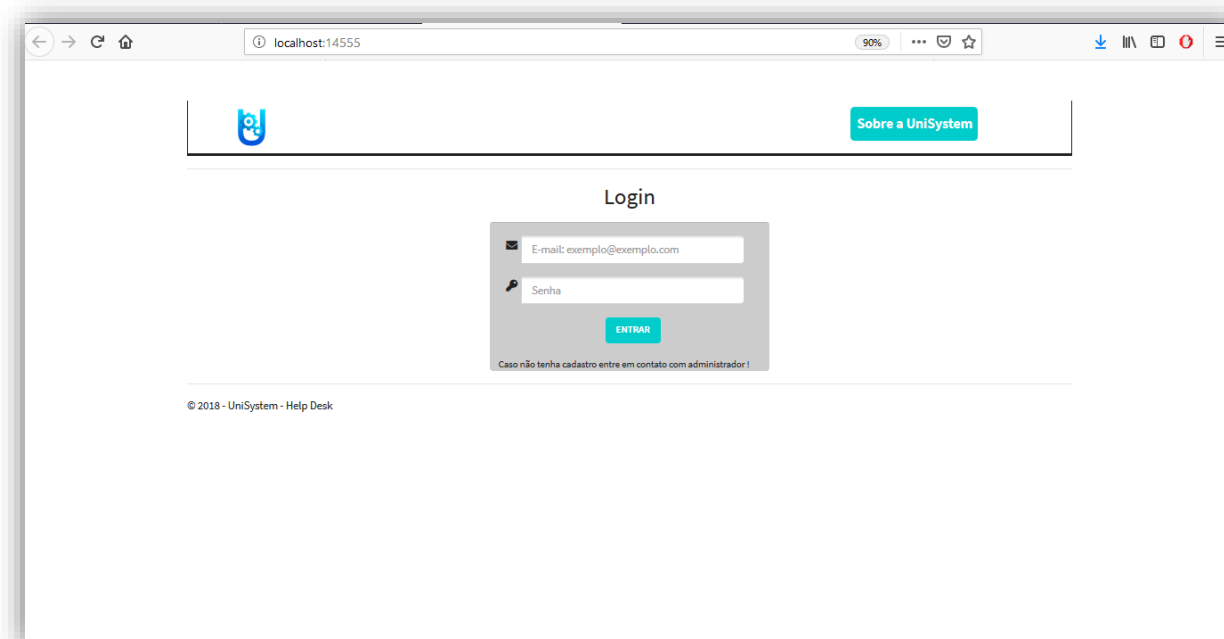
Conforme imagens abaixo:

Imagem 1 – Tela login (Página WEB).



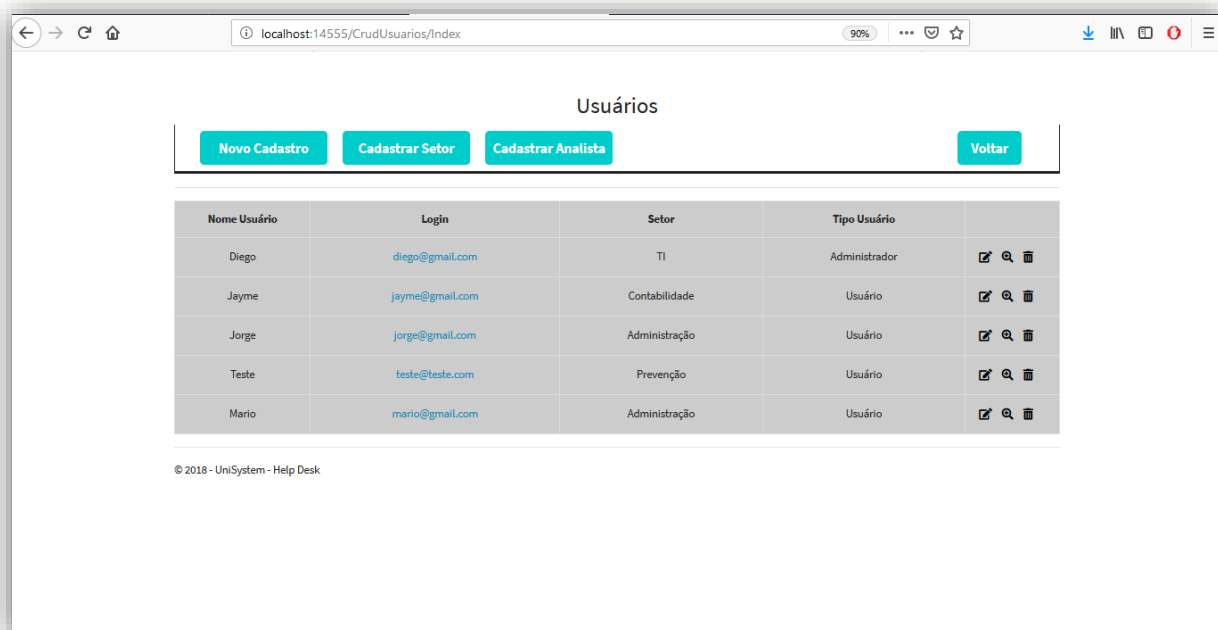
Fonte –Criação própria 2018.

Imagem 2 – Tela inicial Administrador (Página WEB).



Fonte– Criação própria 2018.

Imagem 3 – Tela Consulta de Usuários (Página WEB).



Fonte –Criação própria 2018.

Imagem 4 – Tela Cadastrar Usuário (Página WEB).

Nome

E-mail

Senha

Tipo Usuário

Selecione o tipo de usuário ▼

Setor

Selecione o setor ▼

CADASTRAR

Fonte –Criação própria 2018.

Imagem 5 – Tela Cadastrar Setor (Página WEB).

Cadastrar Setor

Voltar

SETOR

CADASTRAR

© 2018 - UniSystem - Help Desk

Fonte –Criação própria 2018.

Imagem 6 – Tela Cadastrar Analista (Página WEB).

Cadastrar Analista

Voltar

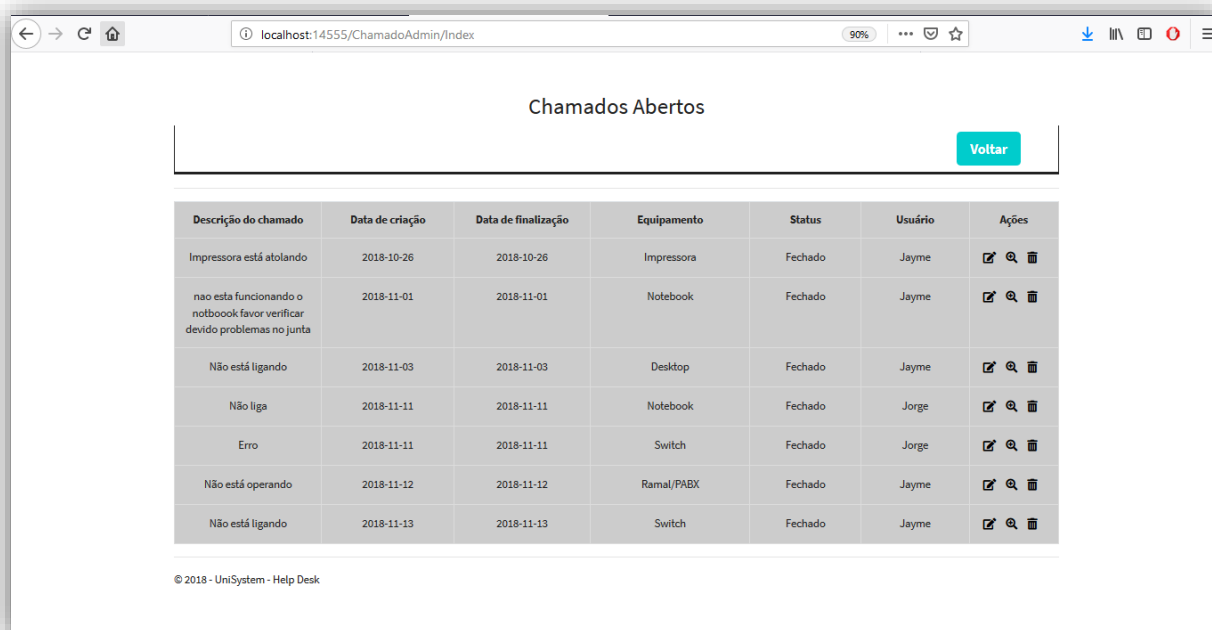
Analista

CADASTRAR

© 2018 - UniSystem - Help Desk

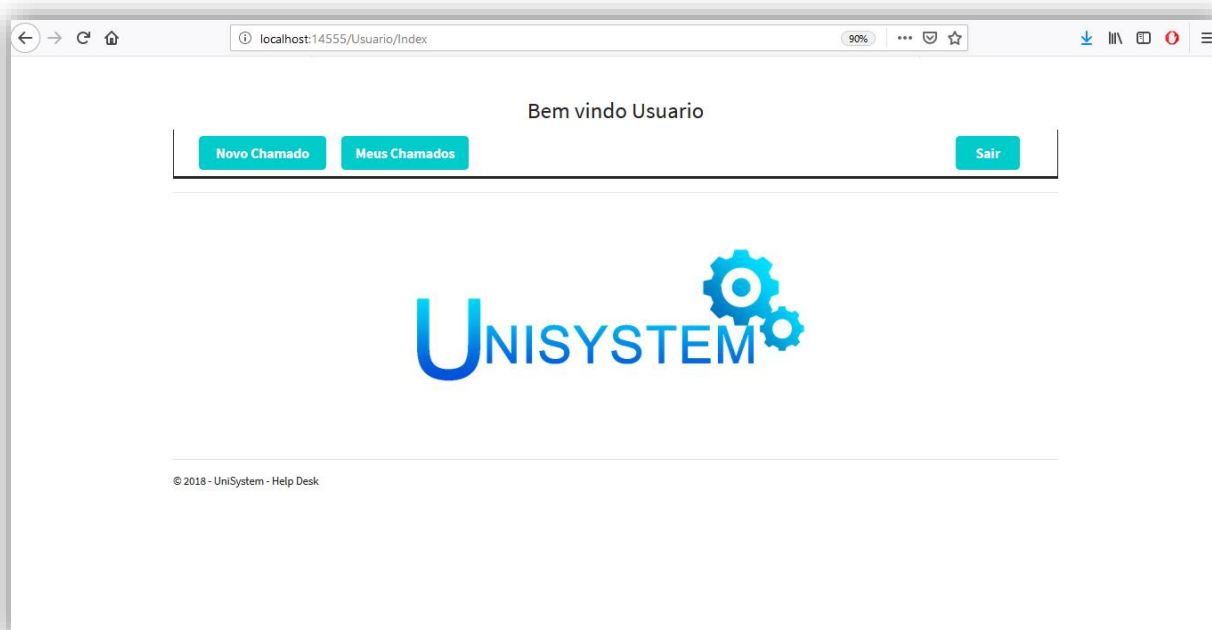
Fonte –Criação própria 2018.

Imagem 7 – Tela Atendimento de Chamado (Página WEB).



Fonte –Criação própria 2018.

Imagem 8 – Tela Inicial Usuário (Página WEB).



Fonte –Criação própria 2018.

Imagem 9 – Tela Abertura de Chamado (Página WEB).

Novo Chamado

Meus Chamados Voltar

Equipamento:
Selecione o Equipamento

Descrição:

Data de Criação
dd / mm / aaaa

CRIAR

© 2018 - UniSystem - Help Desk

Fonte –Criação própria 2018.

Imagem 10 – Tela Abertura de Chamado (Página WEB).

Meus Chamados

Novo Chamado Voltar

Cód chamado	Descrição do chamado	Data de abertura	Equipamento	Status	Ações
1	Impressora está atolando	2018-10-26	Impressora	Fechado	
2	nao esta funcionando o notebook favor verificar devido problemas no junta	2018-11-01	Notebook	Fechado	
3	Não está ligando	2018-11-03	Desktop	Fechado	
9	Não está operando	2018-11-12	Ramal/PABX	Fechado	
11	Não está ligando	2018-11-13	Switch	Fechado	

© 2018 - UniSystem - Help Desk

Fonte –Criação própria 2018.

3.3 Sistema help desk desktop


As funções do sistema baseiam-se em consulta de chamados fechados via web service, apenas os administradores possui acesso ao sistema.

Imagem 1 – Tela Principal (Desktop).



Fonte –Criação própria 2018

Imagem 2 – Tela Consulta (Desktop).



Usuário	Cód chamado	Descrição de chamado	Data abertura	Data finalização	Status	Técnico
Jayme	1	Impressora está atolando	26/10/2018	26/10/2018	Fechado	Diego
Jayme	2	nao esta funcionando o notebook	01/11/2018	01/11/2018	Fechado	Diego
Jayme	3	Não está ligando	03/11/2018	03/11/2018	Fechado	Diego
Jorge	6	Não liga	11/11/2018	11/11/2018	Fechado	Diego
Jorge	8	Erro	11/11/2018	11/11/2018	Fechado	Diego
Jayme	9	Não está operando	12/11/2018	12/11/2018	Fechado	Diego
Jayme	11	Não está ligando	13/11/2018	13/11/2018	Fechado	Diego

Atualizar

Fonte –Criação própria 2018

4.0 FERRAMENTAS UTILIZADAS PARA O DESENVOLVIMENTO

Para o desenvolvimento do sistema help desk foi utilizado as ferramentas:

- Visual Studio
- SQL Server Management Studio

O projeto possui 2 módulos Web e Windows (desktop), o sistema Web contém as quatro operações básicas (CRUD), um layout com formulário simples e tabelas de consulta padronizadas, design com cores suaves e que facilitam a visualização do usuário influenciando diretamente na praticidade de acesso.

4.1 Linguagens e Plataformas Utilizadas

Linguagens de programação são padrões de codificação binária, com sintaxe e semânticas específicas. Desta forma, capazes de criar instruções para máquinas. Graças a esses conjuntos de códigos e recursos, é possível criar programas e sistemas para resolver os mais diversos problemas do cotidiano.

4.2 SQL

A linguagem SQL (Struct Query Language) é uma linguagem padrão para a manipulação de dados dentro de um SGBD. Desenvolvida pela IBM, possibilita o armazenamento, organização, atualização e exclusão de informações dentro do banco de dados.

4.2 ASP .NET

ASP.NET é um framework web da Microsoft, criado com o intuito de permitir aos desenvolvedores criarem páginas web de forma mais rápida e dinâmica é uma linguagem de template, que permite escrever uma página HTML com diretivas de substituição que serão pós-processadas gerando um programa C# ou VB.NET dependendo de como for configurado.

4.3 Windows Presentation Foundation (WPF)

O Windows Presentation Foundation é um componente do Microsoft .NET Framework 3.5. É considerado a próxima geração de UI (User Interface), que permite a criação de aplicações com um grau de personalização extremamente elevado, criando assim uma experiência única ao utilizador.

O WPF suporta interfaces de aplicação, gráficos 2D e 3D, documentos, aceleração de hardware, gráficos vectoriais, visualização de dados interativos e multimídia numa única estrutura (framework).

5.0 MODELO DE PROCESSO DE ENGENHARIA DE SOFTWARE

O modelo de processo de engenharia de software utilizado para o desenvolvimento do sistema de help desk é o modelo de prototipação, pois o foco é compreender e ajudar os clientes e desenvolvedores entender os requisitos para o sistema. Os benefícios da prototipação é a melhoria na facilidade de uso do sistema com uma maior aproximação do sistema com as necessidades dos usuários que possua uma melhoria da qualidade do projeto e na facilidade de manutenção assim reduzindo o esforço no desenvolvimento.

5.1 Objetivos da prototipação evolucionária e descartável

O objetivo da prototipação evolucionária é fornecer aos usuários finais um sistema funcionando, o desenvolvimento começa com aqueles requisitos que são melhores compreendidos.

O objetivo da prototipação descartável é validar ou derivar os requisitos do sistema.

5.2 Pontos negativos da prototipação

O cliente não considera o software, pois pode haver uma confusão do protótipo com a versão final do produto.

Baixa qualidade do desenvolvimento, retrabalho e auto custo de desenvolvimento.

5.3 Pontos positivos da prototipação

Auxilia no entendimento dos requisitos que estão mal compreendidos, pelo cliente e tem uma taxa de desenvolvimento rápido de um projeto.

Ainda que possam ocorrer problemas, a prototipação é um ciclo muito eficiente, o cliente e o desenvolvedor ambos devem concordar que o protótipo seja construído para servir como um mecanismo a fim de definir os requisitos.

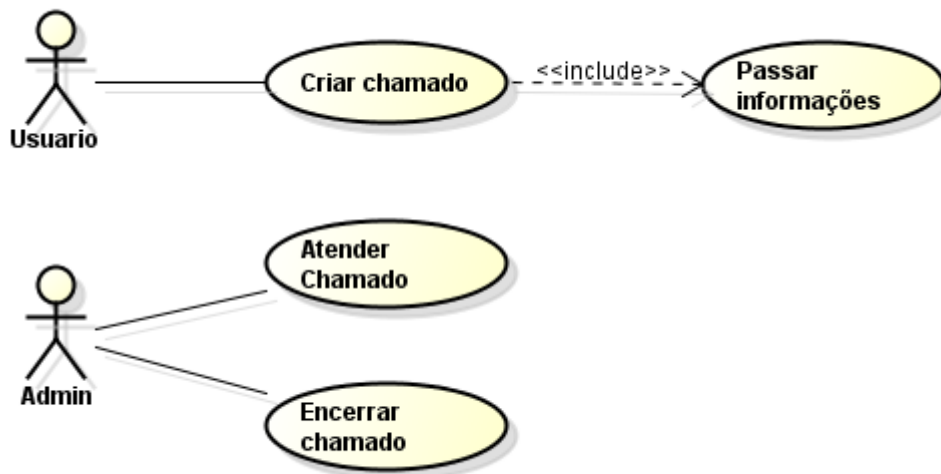
7.0 DIAGRAMA DE CASO DE USO

Diagrama de Casos de Uso tem o objetivo de auxiliar a comunicação entre os analistas e o cliente.

O diagrama de caso de uso descreve um cenário que mostra as funcionalidades do sistema do ponto de vista do usuário é uma excelente ferramenta para o levantamento dos requisitos funcionais do sistema.

O diagrama abaixo representa o caso de uso abertura e atendimento do chamado.

Imagem 1 – Diagrama caso de uso abertura e atendimento chamado.



Fonte: Criação própria, 2018.

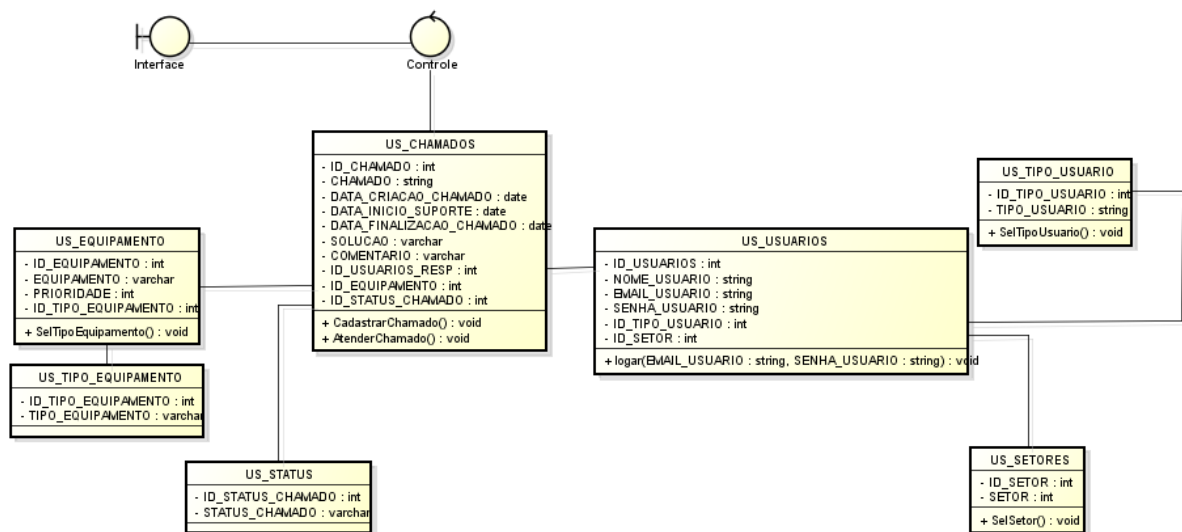
8.0 DIAGRAMA DE CLASSE

Diagrama de classes é tipo de representação é bastante útil no desenvolvimento de sistemas e de softwares de computação, pois define todas as classes que o sistema precisa ter e serve de base para a construção de outros diagramas que definem o tipo de comunicação, sequência e estados dos sistemas.

O diagrama de classes é a parte central da Linguagem de Modelagem Unificada (UML – Unfied Modelling Language). Ele representa as principais finalidades da UML, tendo a função de separar os elementos de design da codificação do sistema.

Diagrama de classe abertura e atendimento de chamado:

Imagem 1 – Diagrama de classe abertura e atendimento chamado.



Fonte: Criação própria, 2018.

9.0 DIAGRAMA DE SEQUÊNCIA

O diagrama de sequência é uma ferramenta importante da UML. Seu principal objetivo é demonstrar em linhas de tempo quais são as interações entre os objetos de um determinado cenário representado pelo diagrama. Geralmente um diagrama de sequência é criado a partir de um diagrama de casos de uso, com a finalidade de descrever como serão as interações/mensagens entre cada objeto/elemento do diagrama. Este diagrama possui dois eixos: o eixo vertical que indica a sequência das mensagens e o tempo de vida dos objetos e o eixo horizontal, este último indica quais são os objetos que participam do diagrama.

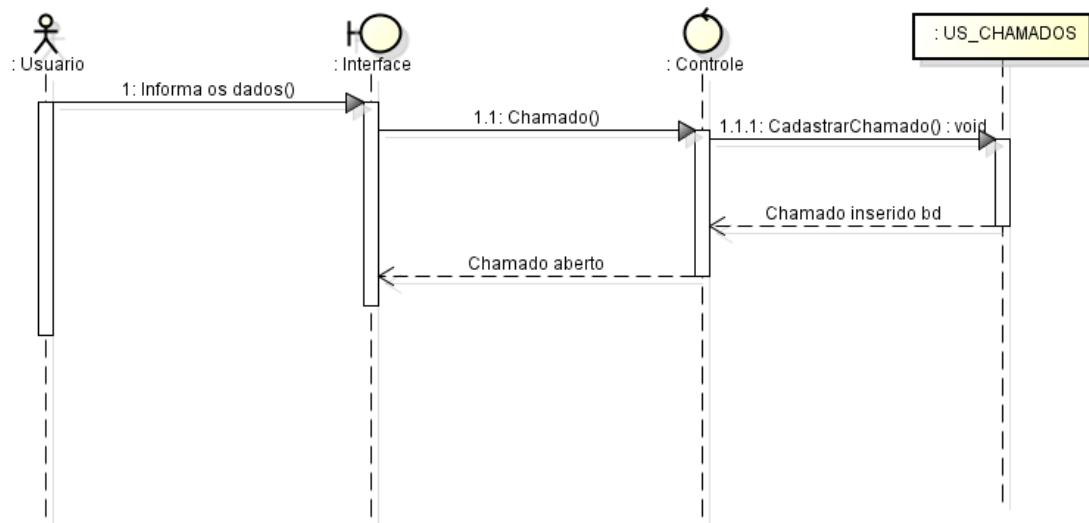
9.1 Elementos Do Diagrama

O diagrama de sequência compõe-se basicamente de dois elementos, são eles objetos e mensagens. Em alguns casos pode-se ter um elemento com a representação semelhante ao elemento Ator do diagrama de casos de uso, contudo este elemento é interpretado como um objeto comum neste diagrama.

9.2 Objetos

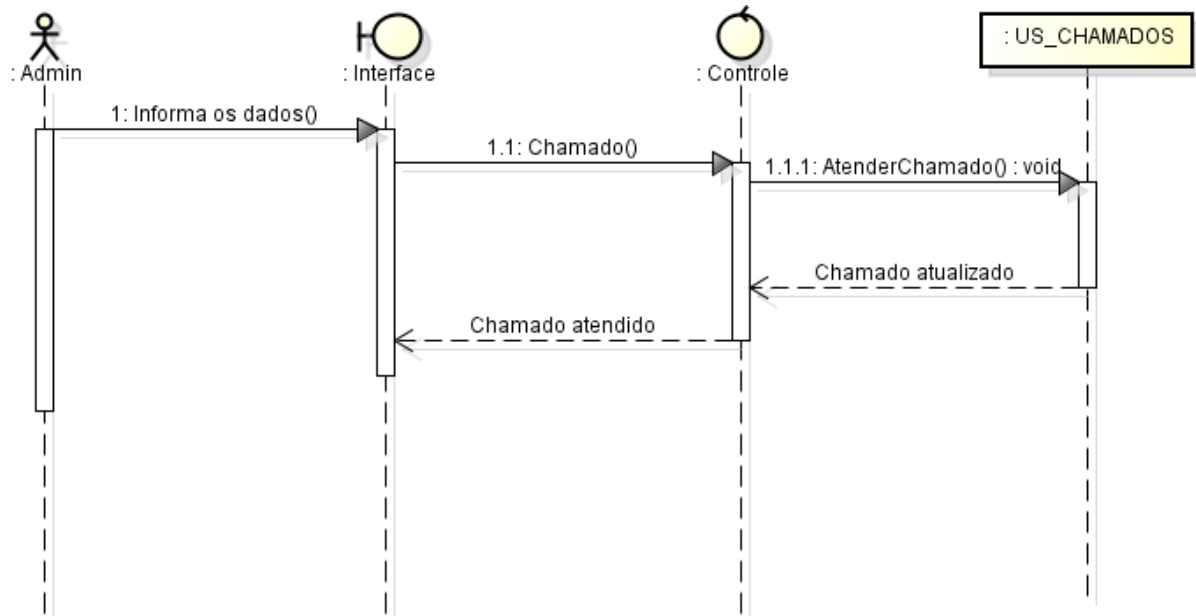
Os objetos podem representar atores ou mesmo instâncias de classes do sistema, desde que tenham alguma interação no cenário do diagrama. Ou seja, geralmente, um objeto vem predefinido de um diagrama de classes. A representação gráfica de um objeto é um retângulo, e a sua distribuição no diagrama se dá pela horizontal. Ou seja, cada objeto do diagrama deve estar alinhado horizontalmente no cenário. A ordem dos objetos não interfere no objetivo do diagrama, porém, é aconselhável distribuir os objetos de forma a facilitar a leitura do diagrama

Imagem 1 – Diagrama de sequência abertura chamado.



Fonte: Criação própria, 2018.

Imagem 2 – Diagrama de sequência atendimento chamado.



Fonte: Criação própria, 2018.

10.0 BANCO DE DADOS

Bancos de dados, são coleções de informações que se relacionam de forma que crie um sentido. São de vital importância para empresas, e há duas décadas se tornaram a principal peça dos sistemas de informação.

O grande objetivo de um banco de dados é oferecer dados abstratos aos usuários. A forma como estes dados estão armazenados e mantidos não interessa aos usuários, mas a disponibilidade eficiente destes dados é que são fundamentais.

Existem vários SGBD (Sistema Gerenciador de Banco de Dados), no mercado, alguns são pagos, outros gratuitos, possui programas que criam uma interface gráfica, gerando um código em SQL automaticamente.

Alguns SGBD:

- SQLServer: Um dos maiores SGBD do mundo, sob licença da Microsoft, tem versões pagas e gratuitas.
- MySQL: O MySQL é um software livre, com código fonte aberto e uso gratuito.
- FirebirdSQL: Roda na maioria dos sistemas Unix, e tem código fonte aberto.
- mSQL: Criado pela Hughes Technologies Pty Ltd., trabalha mais com o uso eficiente da memória, e é um sistema pequeno. Sua licença é altamente controlada pela empresa dona do produto.
- Microsoft Access: é um SGBD da Microsoft que acompanha o pacote Office. É muito utilizado para a aprendizagem e tem poucas atribuições profissionais, devido a sua limitação muito grande em armazenamento.

Para armazenar um dado, é necessário criar tabelas, dentro das tabelas são criadas colunas, onde as informações são armazenadas. Para os dados da base de dados ficarem organizados, devem ser criadas tabelas que não misturem informações.

10.1 Modelo conceitual

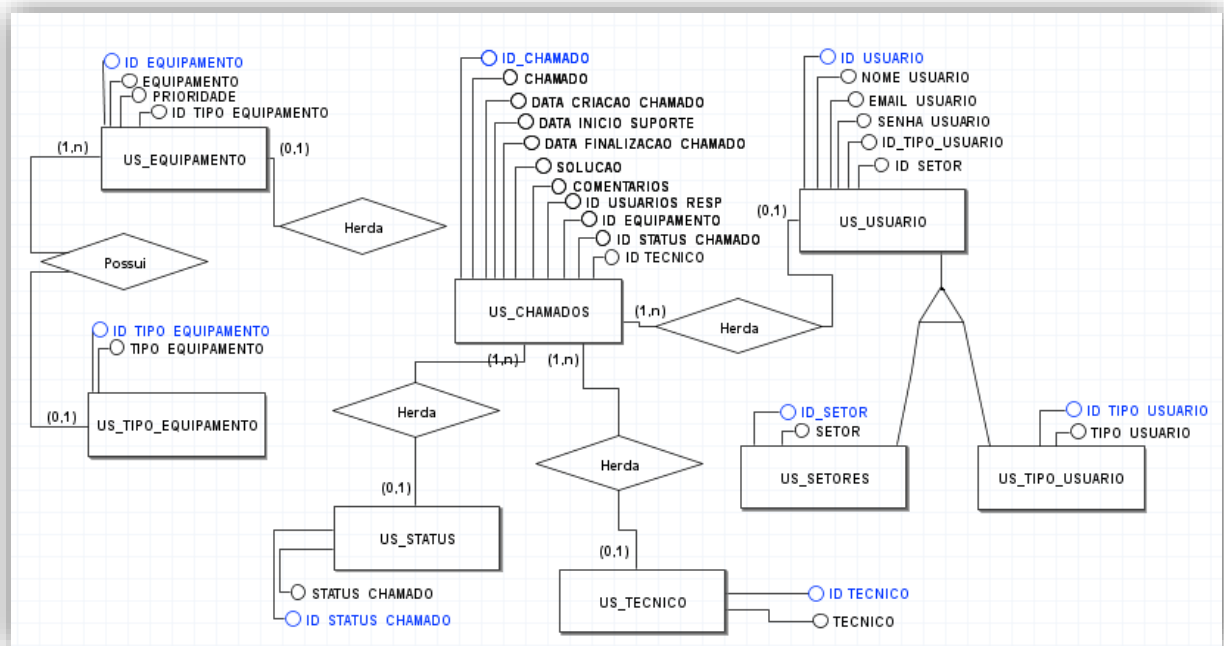
Refere-se ao desenvolvimento de um modelo inicial da base de dados que reflitam as necessidades do usuário.

Essa modelagem preocupa-se em descrever quais dados serão armazenados na base de dados e quais dados se relacionam.

Para fazer o modelo conceitual é necessário entender que dados o usuário final espera que o sistema disponibilize.

A modelagem conceitual fornece uma visão mais próxima do modo como os usuários visualizam os dados realmente.

Imagem 1 – Esquema conceitual banco de dados.



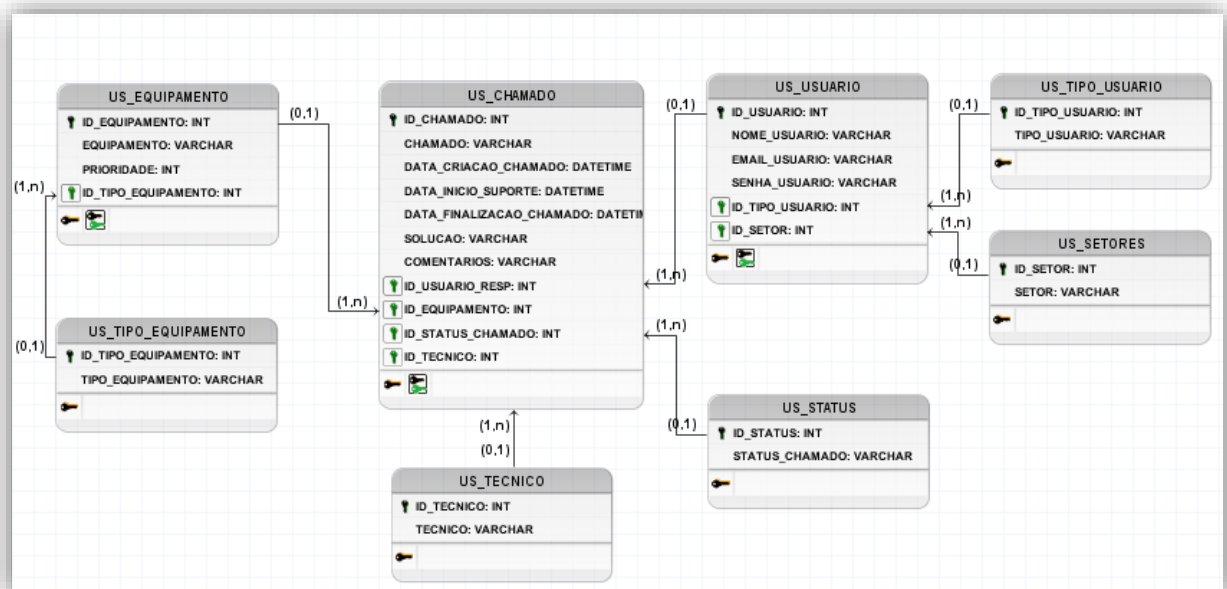
Fonte: Criação própria, 2018.

10.2 Modelo lógico

O projeto lógico consiste no mapeamento do esquema conceitual para o modelo de dados do SGBD adotado.

Isso significa transformar o modelo conceitual obtido na primeira fase num modelo mais próximo da implementação, ou seja, fornece uma visão mais detalhada do modo como os dados são armazenados no computador.

Imagem 1 – Esquema lógico banco de dados.



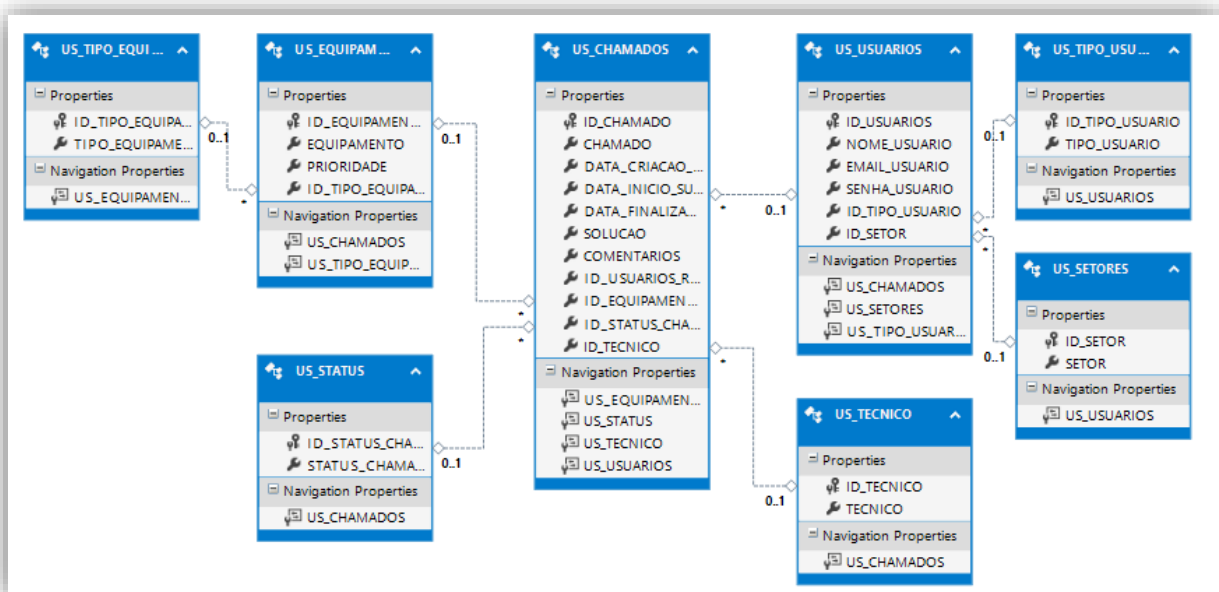
Fonte: Criação própria, 2018.

10.3 Modelo físico

O projeto físico inicia-se a partir do modelo lógico e resulta em um modelo físico. O armazenamento físico do banco de dados, o nível mais baixo de abstração, como os dados estão realmente armazenados.

Um esquema físico é uma descrição da implementação do bando de dados e é direcionado para um SGBD específico.

Imagem 1 – Esquema físico banco de dados.



Fonte: Criação própria, 2018.

11.0 CONCLUSÃO

Os métodos de atendimento do departamento de TI aos usuários do supermercado eram ultrapassados de modo que qualquer problema relacionado ao TI, possuía grandes dificuldades nos atendimentos, pois não tinham controle dos problemas reportados e a falta de comunicação era constante, assim atrasando o suporte e afetando o desempenho operacional do supermercado.

Com a implantação do sistema help desk, a equipe de TI obteve um melhor desempenho ao atender os chamados, pois, cada usuário possui um cadastro, realizado pelo TI (Administrador), para realizar a abertura de chamados, o analista verifica os chamados abertos e realiza os procedimentos necessários para solucionar o problema e retorna o feedback ao usuário.

Dessa forma os processos de atendimento e a organização são mais eficazes, melhorando a operação do supermercado.

12.0 REFERÊNCIAS

Definição de Help Desk

<https://canaltech.com.br/produtos/O-que-e-help-desk>

Acesso em: 13/11/2018.

WebService

<https://www.youtube.com/watch?v=Y5MiVsN41LU>

Acesso em: 30/10/2018

CRUD (4 Operações básicas)

<https://www.youtube.com/watch?v=96hbyHm2w5w&t=0s&list=PLu7YkgKGQpw-pvltTntc6u-9Y29xbUBB&index=5>

Acesso em: 4/10/2018

Autenticação por tipo de usuário

<https://www.youtube.com/watch?v=OSYy919rG6E&index=9&list=PLu7YkgKGQpw-pvltTntc6u-9Y29xbUBB>

Acesso em: 5/10/2018

Data Annotations (Validações)

<https://www.youtube.com/watch?v=pxetlP4Bvns&index=6&list=PLu7YkgKGQpw-pvltTntc6u-9Y29xbUBB>

Acesso em: 6/10/2018

APÊNDICE

APÊNCIDE A – Código ASP .NET

ContaController

```

namespace UniSystemHelpDesk.Controllers
{
    public class ContaController : Controller
    {
        BDUniSystemHelpDesk db = new BDUniSystemHelpDesk ();
        public ActionResult Login()
        {
            return View();
            ViewBag.Message = "";
        }
        [HttpPost]
        public ActionResult Login(US_USUARIOS log)
        {
            var resultado = db.US_USUARIOS.Where(a => a.EMAIL_USUARIO==log.EMAIL_USUARIO
&& a.SENHA_USUARIO==log.SENHA_USUARIO).ToList();
            if (resultado.Count()>0)
            {
                Session["ID_USUARIOS"] = resultado[0].ID_USUARIOS;
                FormsAuthentication.SetAuthCookie(resultado[0].EMAIL_USUARIO, false);
                //Usuario
                if (resultado[0].ID_TIPO_USUARIO==2)
                {
                    return RedirectToAction("../Admin/Index");
                }
                //Admin
                if (resultado[0].ID_TIPO_USUARIO == 1)
                {
                    return RedirectToAction("../Usuario/Index");
                }
            }
            else
            {
                ViewBag.Message = "Usuário incorreto";
            }
            return View(log);
        }
        public ActionResult Logout()
        {
            Session["ID_USUARIOS"] = 0;
            FormsAuthentication.SignOut();
            return RedirectToAction("Login");
        }
        public JsonResult UsuarioDisponivel(string EMAIL_USUARIO)
        {
            return Json(!db.US_USUARIOS.Any(x => x.EMAIL_USUARIO == EMAIL_USUARIO),
            JsonRequestBehavior.AllowGet);
        }
        public ActionResult sobreunisystem()
        {
            return View();
        }
    }
}

```

View ContaController – Login.cshtml

```
@model UniSystemHelpDesk.Models.US_USUARIOS
@{
    ViewBag.Title = "Login";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
            <div class="img-logoci">
                
            </div>
            <div class="btn-marginr">
                @Html.ActionLink("Sobre a UniSystem", "sobreunisystem", "Conta", new { area = "" }, new {
@class = "navbar-brand btn-new" })
            </div>
        </div>
    </div>
</div>
<hr />
<h2>Login</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <div class="form-f">
            <hr />
            @Html.ValidationSummary(true, "", new { @class = "text-danger" })
            <div class="form-group">
                @Html.LabelFor(model => model.EMAIL_USUARIO, "", htmlAttributes: new { @class =
"control-label col-md-2" })
                <div class="col-md-10">
                    <div class="input-group">
                        <span class="input-group-addon">
                            <i class="fas fa-envelope icons-espas"></i>
                        </span>
                        @Html.EditorFor(model => model.EMAIL_USUARIO, new { htmlAttributes = new {
@class = "form-control", placeholder = "E-mail: exemplo@exemplo.com" } })
                    </div>
                        @Html.ValidationMessageFor(model => model.EMAIL_USUARIO, "", new { @class =
"text-danger" })
                    </div>
                </div>
            </div>

            <div class="form-group">
                @Html.LabelFor(model => model.SENHA_USUARIO, "", htmlAttributes: new { @class =
"control-label col-md-2" })
                <div class="col-md-10">
                    <div class="input-group">
                        <span class="input-group-addon">
                            <i class="fas fa-key icons-espas"></i>
                        </span>
                        @Html.EditorFor(model => model.SENHA_USUARIO, new { htmlAttributes = new {
@class = "form-control", placeholder = "Senha" } })
                    </div>
                        @Html.ValidationMessageFor(model => model.SENHA_USUARIO, "", new { @class =
"text-danger" })
                    </div>
                </div>
            </div>
        </div>
    </div>
}
```



```

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Entrar" class="btn btn-default btn-c btn-entrada" />
            </div>
        </div>
        <div class="form-group">
            @ViewBag.Message
        </div>
        <div class="form-group msg-user">
            Caso não tenha cadastro entre em contato com administrador !
        </div>
    </div>
</div>
}

```

View ContaController – sobreunisystem.cshtml

```

@{
    ViewBag.Title = "sobreunisystem";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">

            </div>
            <div>
                @Html.ActionLink("Voltar", "Login", "Conta", new { area = "" }, new { @class = "navbar-brand btn-newvs" })
            </div>
        </div>
    </div>
</div>
<div class="historia">
    <p class="p-m">O propósito da UniSystem é desenvolver sistemas de help desk, para a melhoria do gerenciamento, controle, organização e agilidade nos processos de suporte das demandas, aprimorando a operação do seu negócio.</p>
    <p class="p-m">Os sistemas são desenvolvidos utilizando linguagem de programação (plataforma .NET) e os dados são armazenados no SQL Server.</p>
</div>
<div class="alilogo">
    <div class="logops">
        
    </div>
</div>

```

AdminController

```

namespace UniSystemHelpDesk.Controllers
{
    public class AdminController : Controller
    {
        [Authorize(Roles = "Administrador")]
        public ActionResult Index()
        {
            return View();
        }
    }
}

```

View AdminController – Index.cshtml

```

 ViewBag.Title = "Index";
 Layout = "~/Views/Shared/_LayoutAdmin.cshtml";
 }
 <h2>Bem vindo Administrador</h2>
 <div class="navbar navbar-inverse navbar-fixed-top">
   <div class="container">
     <div class="navbar-header">
       @Html.ActionLink("Cadastrar Usuário", "Index", "CrudUsuarios", new { area = "" }, new { @class
 = "navbar-brand btn-new" })
       @Html.ActionLink("Chamados", "Index", "ChamadoAdmin/Index", new { area = "" }, new {
 @class = "navbar-brand btn-new" })
     </div>
     <div>
       @Html.ActionLink("Sair", "Logout", "Conta", new { area = "" }, new { @class = "navbar-brand
 btn-newvs" })
     </div>
   </div>
 </div>
 <hr/>
 <div class="alilogo">
   <div class="logop">
     
   </div>
 </div>

```

CrudUsuariosController

```

 namespace UniSystemHelpDesk.Controllers
 {
   [Authorize(Roles = "Administrador")]
   public class CrudUsuariosController : Controller
   {
     private BDUniSystemHelpDesk db = new BDUniSystemHelpDesk();

     // GET: CrudUsuarios
     public ActionResult Index()
     {
       var uS_USUARIOS = db.US_USUARIOS.Include(u => u.US_SETORES).Include(u =>
 u.US_TIPO_USUARIO);
       return View(uS_USUARIOS.ToList());
     }

     // GET: CrudUsuarios/Details/5
     public ActionResult Details(int? id)
     {
       if (id == null)
       {
         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
       }
       US_USUARIOS uS_USUARIOS = db.US_USUARIOS.Find(id);
       if (uS_USUARIOS == null)
       {
         return HttpNotFound();
       }
       return View(uS_USUARIOS);
     }
   }
 }

```

```

// GET: CrudUsuarios/Create
public ActionResult Create()
{
    ViewBag.ID_SETOR = db.US_SETORES.ToList();
    ViewBag.ID_TIPO_USUARIO = db.US_TIPO_USUARIO.ToList();
    return View();
}

// POST: CrudUsuarios/Create
// To protect from overposting attacks, please enable the specific properties you want to bind to,
for
// more details see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include =
"ID_USUARIOS,NOME_USUARIO,EMAIL_USUARIO,SENHA_USUARIO,ID_TIPO_USUARIO,ID_SE
TOR")] US_USUARIOS uS_USUARIOS)
{
    if (ModelState.IsValid)
    {
        db.US_USUARIOS.Add(uS_USUARIOS);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    ViewBag.ID_SETOR = new SelectList(db.US_SETORES, "ID_SETOR", "SETOR",
uS_USUARIOS.ID_SETOR);
    ViewBag.ID_TIPO_USUARIO = new SelectList(db.US_TIPO_USUARIO,
"ID_TIPO_USUARIO", "TIPO_USUARIO", uS_USUARIOS.ID_TIPO_USUARIO);
    return View(uS_USUARIOS);
}

// GET: CrudUsuarios/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    US_USUARIOS uS_USUARIOS = db.US_USUARIOS.Find(id);
    if (uS_USUARIOS == null)
    {
        return HttpNotFound();
    }
    ViewBag.ID_SETOR = new SelectList(db.US_SETORES, "ID_SETOR", "SETOR",
uS_USUARIOS.ID_SETOR);
    ViewBag.ID_TIPO_USUARIO = new SelectList(db.US_TIPO_USUARIO,
"ID_TIPO_USUARIO", "TIPO_USUARIO", uS_USUARIOS.ID_TIPO_USUARIO);
    return View(uS_USUARIOS);
}

// POST: CrudUsuarios/Edit/5
// To protect from overposting attacks, please enable the specific properties you want to bind to,
for
// more details see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
"ID_USUARIOS,NOME_USUARIO,EMAIL_USUARIO,SENHA_USUARIO,ID_TIPO_USUARIO,ID_SE
TOR")] US_USUARIOS uS_USUARIOS)

```

```

    {
        if (ModelState.IsValid)
        {
            db.Entry(uS_USUARIOS).State = EntityState.Modified;
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        ViewBag.ID_SETOR = new SelectList(db.US_SETORES, "ID_SETOR", "SETOR",
uS_USUARIOS.ID_SETOR);
        ViewBag.ID_TIPO_USUARIO = new SelectList(db.US_TIPO_USUARIO,
"ID_TIPO_USUARIO", "TIPO_USUARIO", uS_USUARIOS.ID_TIPO_USUARIO);
        return View(uS_USUARIOS);
    }

    // GET: CrudUsuarios/Delete/5
    public ActionResult Delete(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        US_USUARIOS uS_USUARIOS = db.US_USUARIOS.Find(id);
        if (uS_USUARIOS == null)
        {
            return HttpNotFound();
        }
        return View(uS_USUARIOS);
    }

    // POST: CrudUsuarios/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public ActionResult DeleteConfirmed(int id)
    {
        US_USUARIOS uS_USUARIOS = db.US_USUARIOS.Find(id);
        db.US_USUARIOS.Remove(uS_USUARIOS);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    {
        if (disposing)
        {
            db.Dispose();
        }
        base.Dispose(disposing);
    }
}

```

View CrudUsuariosController Index.cshtml

```

@model IEnumerable<UniSystemHelpDesk.Models.US_USUARIOS>
@{
    ViewBag.Title = "Index";
}
<h2>Usuários</h2>
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">

```

```

        <div class="navbar-header">
            @Html.ActionLink("Novo Cadastro", "Create", "CrudUsuarios", new { area = "" }, new { @class
= "navbar-brand btn-new" })
            @Html.ActionLink("Cadastrar Setor", "Create", "AddSetor", new { area = "" }, new { @class =
"navbar-brand btn-new" })
            @Html.ActionLink("Cadastrar Analista", "Create", "AddTecnico", new { area = "" }, new { @class
= "navbar-brand btn-new" })
        </div>
        <div>
            @Html.ActionLink("Voltar", "Index", "Admin", new { area = "" }, new { @class = "navbar-brand
btn-newvs" })
        </div>
    </div>
</div>
<hr />
<table class="table table-l">
    <tr>
        <th>
            @Html.DisplayName("Nome Usuário")
        </th>
        <th>
            @Html.DisplayName("Login")
        </th>
        <th>
            @Html.DisplayName("Setor")
        </th>
        <th>
            @Html.DisplayName("Tipo Usuário")
        </th>
    </tr>
    @foreach (var item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.NOME_USUARIO)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.EMAIL_USUARIO)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.US_SETORES.SETOR)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.US_TIPO_USUARIO.TIPO_USUARIO)
            </td>
            <td class="ed-link">
                <a href="@Url.Action("Edit", "CrudUsuarios", new { id = item.ID_USUARIOS, @title = "Editar"
})" class="fas fa-edit icons-esp link-hover" title="Editar"></a>
                <a href="@Url.Action("Details", "CrudUsuarios", new { id = item.ID_USUARIOS, @title =
"Detalhes" })" class="fas fa-search-plus icons-esp link-hover" title="Detalhes"></a>
                <a href="@Url.Action("Delete", "CrudUsuarios", new { id = item.ID_USUARIOS, @title =
"Deletar" })" class="fas fa-trash-alt icons-esp link-hover" title="Deletar"></a>
            </td>
        </tr>
    }
</table>

```

View CrudUsuariosController Edit.cshtml

```

@model UniSystemHelpDesk.Models.US_USUARIOS
@{
    ViewBag.Title = "Edit";
}
<h2>Editar Usuário</h2>
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
            </div>
        </div>
        @Html.ActionLink("Voltar", "Index", "CrudUsuarios", new { area = "" }, new { @class = "navbar-brand btn-newvs" })
    </div>
</div>
<hr />
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal form-alin">
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.ID_USUARIOS)
        <div class="form-group">
            @Html.LabelFor(model => model.NOME_USUARIO, "Nome", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.NOME_USUARIO, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.NOME_USUARIO, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.EMAIL_USUARIO, "E-mail", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.EMAIL_USUARIO, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.EMAIL_USUARIO, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.SENHA_USUARIO, "Senha", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.SENHA_USUARIO, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.SENHA_USUARIO, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.ID_TIPO_USUARIO, "Tipo Usuário", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">

```

```

        @Html.DropDownList("ID_TIPO_USUARIO", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.ID_TIPO_USUARIO, "", new { @class = "text-danger" })
    </div>
</div>
<div class="form-group">
    @Html.LabelFor(model => model.ID_SETOR, "Setor", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("ID_SETOR", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.ID_SETOR, "", new { @class = "text-danger" })
    </div>
</div>
<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Editar" class="btn btn-default btn-l btn-entrada" />
    </div>
</div>
</div>
}
<script src="~/Scripts/jquery-1.10.2.min.js"></script>
<script src="~/Scripts/jquery.validate.min.js"></script>
<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>

```

View CrudUsuariosController Details.cshtml

```

@model UniSystemHelpDesk.Models.US_USUARIOS
@{
    ViewBag.Title = "Details";
}
<h2>Detalhes Usuários</h2>
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
            @Html.ActionLink("Editar", "Edit", new { id = Model.ID_USUARIOS }, new { @class = "navbar-brand btn-new" })
        </div>
        <div>
            @Html.ActionLink("Voltar", "Index", "CrudUsuarios", new { area = "" }, new { @class = "navbar-brand btn-newvs" })
        </div>
    </div>
</div>
<div>
    <hr />
    <dl class="dl-horizontal form-alin2">
        <dt>
            @Html.DisplayName("NOME")
        </dt>
        <dd>
            @Html.DisplayFor(model => model.NOME_USUARIO)
        </dd>
        <dt>
            @Html.DisplayName("EMAIL")
        </dt>
        <dd>
            @Html.DisplayFor(model => model.EMAIL_USUARIO)
        </dd>
    </dl>

```

```

<dt>
    @Html.DisplayName("SETOR")
</dt>
<dd>
    @Html.DisplayFor(model => model.US_SETORES.SETOR)
</dd>
<dt>
    @Html.DisplayName("TIPO USUARIO")
</dt>
<dd>
    @Html.DisplayFor(model => model.US_TIPO_USUARIO.TIPO_USUARIO)
</dd>
</dl>
</div>

```

View CrudUsuariosController Delete.cshtml

```

@model UniSystemHelpDesk.Models.US_USUARIOS
@{
    ViewBag.Title = "Delete";
}
<h2>Deletar Usuário</h2>
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">

            </div>
            <div>
                @Html.ActionLink("Voltar", "Index", "CrudUsuarios", new { area = "" }, new { @class = "navbar-
brand btn-newvs" })
            </div>
        </div>
    </div>
</div>
<hr />
<h3>Tem certeza que deseja excluir esse usuário ?</h3>
<div>
    <hr />
    <dl class="dl-horizontal form-alin2">
        <dt>
            @Html.DisplayName("NOME")
        </dt>
        <dd>
            @Html.DisplayFor(model => model.NOME_USUARIO)
        </dd>
        <dt>
            @Html.DisplayName("EMAIL")
        </dt>
        <dd>
            @Html.DisplayFor(model => model.EMAIL_USUARIO)
        </dd>
        <dt>
            @Html.DisplayName("SETOR")
        </dt>
        <dd>
            @Html.DisplayFor(model => model.US_SETORES.SETOR)
        </dd>
        <dt>
            @Html.DisplayName("TIPO USUARIO")
        </dt>
        <dd>

```



```

        @Html.DisplayFor(model => model.US_TIPO_USUARIO.TIPO_USUARIO)
    </dd>
</dl>
@using (Html.BeginForm()) {
    @Html.AntiForgeryToken()

    <div class="form-actions no-color">
        <input type="submit" value="Deletar" class="btn btn-default btn-d" />
    </div>
}
</div>

```

View CrudUsuariosController Create.cshtml

```

@model UniSystemHelpDesk.Models.US_USUARIOS
@{
    ViewBag.Title = "Create";
}
<h2>Cadastrar Usuários</h2>
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
        </div>
        <div>
            @Html.ActionLink("Voltar", "Index", "CrudUsuarios", new { area = "" }, new { @class = "navbar-brand btn-newvs" })
        </div>
    </div>
</div>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <hr />
    <div class="form-horizontal form-alin">
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.NOME_USUARIO, "Nome", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.NOME_USUARIO, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.NOME_USUARIO, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.EMAIL_USUARIO, "E-mail", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.EMAIL_USUARIO, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.EMAIL_USUARIO, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.SENHA_USUARIO, "Senha", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">

```

```

        @Html.EditorFor(model => model.SENHA_USUARIO, new { htmlAttributes = new { @class =
"form-control" } })
        @Html.ValidationMessageFor(model => model.SENHA_USUARIO, "", new { @class = "text-
danger" })
    </div>
</div>
<div class="form-group">
    @Html.LabelFor(model => model.ID_TIPO_USUARIO, "Tipo Usuário", htmlAttributes: new {
@class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("ID_TIPO_USUARIO", new SelectList(ViewBag.ID_TIPO_USUARIO,
"ID_TIPO_USUARIO", "TIPO_USUARIO"), "Selecione o tipo de usuário", htmlAttributes: new { @class
= "form-control" })
        @Html.ValidationMessageFor(model => model.ID_TIPO_USUARIO, "", new { @class = "text-
danger" })
    </div>
</div>
<div class="form-group">
    @Html.LabelFor(model => model.ID_SETOR, "Setor", htmlAttributes: new { @class = "control-
label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("ID_SETOR", new SelectList(ViewBag.ID_SETOR, "ID_SETOR",
"SETOR"), "Selecione o setor", htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.ID_SETOR, "", new { @class = "text-danger"
})
    </div>
</div>
<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Cadastrar" class="btn btn-default btn-l btn-entrada" />
    </div>
</div>
</div>
}
</div>
</div>
<script src="~/Scripts/jquery-1.10.2.min.js"></script>
<script src="~/Scripts/jquery.validate.min.js"></script>
<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>

```

AddSetorController

```

namespace UniSystemHelpDesk.Controllers
{
    [Authorize(Roles = "Administrador")]
    public class AddSetorController : Controller
    {
        private BDUniSystemHelpDesk db = new BDUniSystemHelpDesk();

        // GET: AddSetor
        public ActionResult Index()
        {
            return View(db.US_SETORES.ToList());
        }

        // GET: AddSetor/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {

```

```

        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    US_SETORES uS_SETORES = db.US_SETORES.Find(id);
    if (uS_SETORES == null)
    {
        return HttpNotFound();
    }
    return View(uS_SETORES);
}

// GET: AddSetor/Create
public ActionResult Create()
{
    return View();
}

// POST: AddSetor/Create
// To protect from overposting attacks, please enable the specific properties you want to bind to,
for
// more details see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "ID_SETOR,SETOR")] US_SETORES uS_SETORES)
{
    if (ModelState.IsValid)
    {
        db.US_SETORES.Add(uS_SETORES);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    return View(uS_SETORES);
}

// GET: AddSetor/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    US_SETORES uS_SETORES = db.US_SETORES.Find(id);
    if (uS_SETORES == null)
    {
        return HttpNotFound();
    }
    return View(uS_SETORES);
}

// POST: AddSetor/Edit/5
// To protect from overposting attacks, please enable the specific properties you want to bind to,
for
// more details see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include = "ID_SETOR,SETOR")] US_SETORES uS_SETORES)
{
    if (ModelState.IsValid)
    {
        db.Entry(uS_SETORES).State = EntityState.Modified;
    }
}

```

```

        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(uS_SETORES);
}

// GET: AddSetor/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    US_SETORES uS_SETORES = db.US_SETORES.Find(id);
    if (uS_SETORES == null)
    {
        return HttpNotFound();
    }
    return View(uS_SETORES);
}

// POST: AddSetor/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    US_SETORES uS_SETORES = db.US_SETORES.Find(id);
    db.US_SETORES.Remove(uS_SETORES);
    db.SaveChanges();
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}
}

```

View AddSetorController Create.cshtml

```

@model UniSystemHelpDesk.Models.US_SETORES
@{
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Cadastrar Setor</h2>
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
            </div>
        <div>
            @Html.ActionLink("Voltar", "Index", "CrudUsuarios", new { area = "" }, new { @class = "navbar-brand btn-newvs" })
        </div>
    </div>
</div>

```

```

</div>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <hr />
    <div class="form-horizontal form-alin">
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.SETOR, htmlAttributes: new { @class = "control-label col-md-
2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.SETOR, new { htmlAttributes = new { @class = "form-
control" } })
                @Html.ValidationMessageFor(model => model.SETOR, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Cadastrar" class="btn btn-default btn-l btn-entrada" />
            </div>
        </div>
    </div>
}
<script src="~/Scripts/jquery-1.10.2.min.js"></script>
<script src="~/Scripts/jquery.validate.min.js"></script>
<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>

```

AddTecnicoController

```

namespace UniSystemHelpDesk.Controllers
{
    [Authorize(Roles = "Administrador")]
    public class AddTecnicoController : Controller
    {
        private BDUiSystemHelpDesk db = new BDUiSystemHelpDesk();

        // GET: AddTecnico
        public ActionResult Index()
        {
            return View(db.US_TECNICO.ToList());
        }

        // GET: AddTecnico/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            US_TECNICO uS_TECNICO = db.US_TECNICO.Find(id);
            if (uS_TECNICO == null)
            {
                return HttpNotFound();
            }
            return View(uS_TECNICO);
        }

        // GET: AddTecnico/Create
        public ActionResult Create()
        {

```

```

        return View();
    }

    // POST: AddTecnico/Create
    // To protect from overposting attacks, please enable the specific properties you want to bind to,
for
    // more details see http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Create([Bind(Include = "ID_TECNICO,TECNICO")] US_TECNICO
uS_TECNICO)
    {
        if (ModelState.IsValid)
        {
            db.US_TECNICO.Add(uS_TECNICO);
            db.SaveChanges();
            return RedirectToAction("Index");
        }

        return View(uS_TECNICO);
    }

    // GET: AddTecnico/Edit/5
    public ActionResult Edit(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        US_TECNICO uS_TECNICO = db.US_TECNICO.Find(id);
        if (uS_TECNICO == null)
        {
            return HttpNotFound();
        }
        return View(uS_TECNICO);
    }

    // POST: AddTecnico/Edit/5
    // To protect from overposting attacks, please enable the specific properties you want to bind to,
for
    // more details see http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Edit([Bind(Include = "ID_TECNICO,TECNICO")] US_TECNICO
uS_TECNICO)
    {
        if (ModelState.IsValid)
        {
            db.Entry(uS_TECNICO).State = EntityState.Modified;
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        return View(uS_TECNICO);
    }

    // GET: AddTecnico/Delete/5
    public ActionResult Delete(int? id)
    {
        if (id == null)
        {

```

```

        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    US_TECNICO uS_TECNICO = db.US_TECNICO.Find(id);
    if (uS_TECNICO == null)
    {
        return HttpNotFound();
    }
    return View(uS_TECNICO);
}

// POST: AddTecnico/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    US_TECNICO uS_TECNICO = db.US_TECNICO.Find(id);
    db.US_TECNICO.Remove(uS_TECNICO);
    db.SaveChanges();
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}
}

```

View AddTecnicoController Create.cshtml

```

@model UniSystemHelpDesk.Models.US_TECNICO
@{
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Cadastrar Analista</h2>
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
            </div>
        <div>
            @Html.ActionLink("Voltar", "Index", "CrudUsuarios", new { area = "" }, new { @class = "navbar-brand btn-newvs" })
        </div>
    </div>
</div>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <hr />
    <div class="form-horizontal form-alin">
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.TECNICO, "Analista", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">

```

```

        @Html.EditorFor(model => model.TECNICO, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.TECNICO, "", new { @class = "text-danger"
    })
    </div>
</div>
<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Cadastrar" class="btn btn-default btn-l btn-entrada" />
    </div>
</div>
</div>
}
<script src="~/Scripts/jquery-1.10.2.min.js"></script>
<script src="~/Scripts/jquery.validate.min.js"></script>
<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>

```

ChamadoAdminController

```

namespace UniSystemHelpDesk.Controllers
{
    [Authorize(Roles = "Administrador")]
    public class ChamadoAdminController : Controller
    {
        private BDUniSystemHelpDesk db = new BDUniSystemHelpDesk();

        // GET: ChamadoAdmin
        public ActionResult Index()
        {
            var uS_CHAMADOS = db.US_CHAMADOS.Include(u => u.US_EQUIPAMENTO).Include(u =>
u.US_STATUS).Include(u => u.US_TECNICO).Include(u => u.US_USUARIOS);
            return View(uS_CHAMADOS.ToList());
        }

        // GET: ChamadoAdmin/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            US_CHAMADOS uS_CHAMADOS = db.US_CHAMADOS.Find(id);
            if (uS_CHAMADOS == null)
            {
                return HttpNotFound();
            }
            return View(uS_CHAMADOS);
        }

        // GET: ChamadoAdmin/Create
        public ActionResult Create()
        {
            ViewBag.ID_EQUIPAMENTO = new SelectList(db.US_EQUIPAMENTO,
"ID_EQUIPAMENTO", "EQUIPAMENTO");
            ViewBag.ID_STATUS_CHAMADO = new SelectList(db.US_STATUS,
"ID_STATUS_CHAMADO", "STATUS_CHAMADO");
            ViewBag.ID_TECNICO = new SelectList(db.US_TECNICO, "ID_TECNICO", "TECNICO");
            ViewBag.ID_USUARIOS_RESP = new SelectList(db.US_USUARIOS, "ID_USUARIOS",
"NOME_USUARIO");
        }
    }
}

```



```

        return View();
    }

    // POST: ChamadoAdmin/Create
    // To protect from overposting attacks, please enable the specific properties you want to bind to,
    for
    // more details see http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Create([Bind(Include =
    "ID_CHAMADO,CHAMADO,DATA_CRIACAO_CHAMADO,DATA_INICIO_SUPOORTE,DATA_FINALIZ
    Acao_CHAMADO,SOLUCAO,COMENTARIOS,ID_USUARIOS_RESP,ID_EQUIPAMENTO,ID_STAT
    US_CHAMADO,ID_TECNICO")] US_CHAMADOS uS_CHAMADOS)
    {
        if (ModelState.IsValid)
        {
            db.US_CHAMADOS.Add(uS_CHAMADOS);
            db.SaveChanges();
            return RedirectToAction("Index");
        }

        ViewBag.ID_EQUIPAMENTO = new SelectList(db.US_EQUIPAMENTO,
        "ID_EQUIPAMENTO", "EQUIPAMENTO", uS_CHAMADOS.ID_EQUIPAMENTO);
        ViewBag.ID_STATUS_CHAMADO = new SelectList(db.US_STATUS,
        "ID_STATUS_CHAMADO", "STATUS_CHAMADO", uS_CHAMADOS.ID_STATUS_CHAMADO);
        ViewBag.ID_TECNICO = new SelectList(db.US_TECNICO, "ID_TECNICO", "TECNICO",
        uS_CHAMADOS.ID_TECNICO);
        ViewBag.ID_USUARIOS_RESP = new SelectList(db.US_USUARIOS, "ID_USUARIOS",
        "NOME_USUARIO", uS_CHAMADOS.ID_USUARIOS_RESP);
        return View(uS_CHAMADOS);
    }

    // GET: ChamadoAdmin/Edit/5
    public ActionResult Edit(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        US_CHAMADOS uS_CHAMADOS = db.US_CHAMADOS.Find(id);
        if (uS_CHAMADOS == null)
        {
            return HttpNotFound();
        }
        ViewBag.ID_EQUIPAMENTO = new SelectList(db.US_EQUIPAMENTO,
        "ID_EQUIPAMENTO", "EQUIPAMENTO", uS_CHAMADOS.ID_EQUIPAMENTO);
        ViewBag.ID_STATUS_CHAMADO = new SelectList(db.US_STATUS,
        "ID_STATUS_CHAMADO", "STATUS_CHAMADO", uS_CHAMADOS.ID_STATUS_CHAMADO);
        ViewBag.ID_TECNICO = new SelectList(db.US_TECNICO, "ID_TECNICO", "TECNICO",
        uS_CHAMADOS.ID_TECNICO);
        ViewBag.ID_USUARIOS_RESP = new SelectList(db.US_USUARIOS, "ID_USUARIOS",
        "NOME_USUARIO", uS_CHAMADOS.ID_USUARIOS_RESP);
        return View(uS_CHAMADOS);
    }

    // POST: ChamadoAdmin/Edit/5
    // To protect from overposting attacks, please enable the specific properties you want to bind to,
    for
    // more details see http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]

```

```

[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
"ID_CHAMADO,CHAMADO,DATA_CRIACAO_CHAMADO,DATA_INICIO_SUPORE,DATA_FINALIZ
ACAO_CHAMADO,SOLUCAO,COMENTARIOS,ID_USUARIOS_RESP,ID_EQUIPAMENTO,ID_STAT
US_CHAMADO,ID_TECNICO")] US_CHAMADOS uS_CHAMADOS)
{
    if (ModelState.IsValid)
    {
        db.Entry(uS_CHAMADOS).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    ViewBag.ID_EQUIPAMENTO = new SelectList(db.US_EQUIPAMENTO,
"ID_EQUIPAMENTO", "EQUIPAMENTO", uS_CHAMADOS.ID_EQUIPAMENTO);
    ViewBag.ID_STATUS_CHAMADO = new SelectList(db.US_STATUS,
"ID_STATUS_CHAMADO", "STATUS_CHAMADO", uS_CHAMADOS.ID_STATUS_CHAMADO);
    ViewBag.ID_TECNICO = new SelectList(db.US_TECNICO, "ID_TECNICO", "TECNICO",
uS_CHAMADOS.ID_TECNICO);
    ViewBag.ID_USUARIOS_RESP = new SelectList(db.US_USUARIOS, "ID_USUARIOS",
"NOME_USUARIO", uS_CHAMADOS.ID_USUARIOS_RESP);
    return View(uS_CHAMADOS);
}

// GET: ChamadoAdmin/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    US_CHAMADOS uS_CHAMADOS = db.US_CHAMADOS.Find(id);
    if (uS_CHAMADOS == null)
    {
        return HttpNotFound();
    }
    return View(uS_CHAMADOS);
}

// POST: ChamadoAdmin/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    US_CHAMADOS uS_CHAMADOS = db.US_CHAMADOS.Find(id);
    db.US_CHAMADOS.Remove(uS_CHAMADOS);
    db.SaveChanges();
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}
}

```

View ChamadoAdminController Index.cshtml

```
@model IEnumerable<UniSystemHelpDesk.Models.US_CHAMADOS>
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Chamados Abertos</h2>

<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">

            </div>
            <div>
                @Html.ActionLink("Voltar", "Index", "Admin", new { area = "" }, new { @class = "navbar-brand
btn-newvs" })
            </div>
        </div>
    </div>
</div>
<hr />
<table class="table table-l">
    <tr>
        <th>
            @Html.DisplayName("Descrição do chamado")
        </th>
        <th>
            @Html.DisplayName("Data de criação")
        </th>
        <th>
            @Html.DisplayName("Data de finalização")
        </th>
        <th>
            @Html.DisplayName("Equipamento")
        </th>
        <th>
            @Html.DisplayName("Status")
        </th>
        <th>
            @Html.DisplayName("Usuário")
        </th>
        <th>
            Ações
        </th>
    </tr>
    @foreach (var item in Model) {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.CHAMADO)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.DATA_CRIACAO_CHAMADO)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.DATA_FINALIZACAO_CHAMADO)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.US_EQUIPAMENTO.EQUIPAMENTO)
            </td>
            <td>
            </td>
        </tr>
    }
```

```

        @Html.DisplayFor(modelItem => item.US_STATUS.STATUS_CHAMADO)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.US_USUARIOS.NOME_USUARIO)
    </td>
    <td class="ed-link teste">
        <a href="@Url.Action("Edit", "ChamadoAdmin", new{ id = item.ID_CHAMADO, @title="Atender"
        })" class="fas fa-edit icons-espas link-hover" title="Atender"></a>
        <a href="@Url.Action("Details", "ChamadoAdmin", new { id = item.ID_CHAMADO, @title =
        "Detalhes" })" class="fas fa-search-plus icons-espas link-hover" title="Detalhes"></a>
        <a href="@Url.Action("Delete", "ChamadoAdmin", new { id = item.ID_CHAMADO, @title =
        "Excluir" })" class="fas fa-trash-alt icons-espas link-hover" title="Excluir"></a>
    </td>
</tr>
}
</table>

```

View ChamadoAdminController Edit.cshtml

```

@model UniSystemHelpDesk.Models.US_CHAMADOS
@{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Atender Chamado</h2>
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
        </div>
        <div>
            @Html.ActionLink("Voltar", "Index", "ChamadoAdmin", new { area = "" }, new { @class =
            "navbar-brand btn-newvs" })
        </div>
    </div>
</div>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <hr />
    <div class="form-horizontal form-alin">
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.ID_CHAMADO)

        <div class="form-group">
            @Html.LabelFor(model => model.ID_EQUIPAMENTO, "Equipamento", htmlAttributes: new {
            @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("ID_EQUIPAMENTO", null, htmlAttributes: new { @class = "form-
                control", @readonly = "readonly" })
                @Html.ValidationMessageFor(model => model.ID_EQUIPAMENTO, "", new { @class = "text-
                danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.CHAMADO, "Descrição", htmlAttributes: new { @class =
            "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.CHAMADO, new { htmlAttributes = new { @class = "form-
                control", @readonly = "readonly" } })
            </div>
        </div>
    </div>
}

```

```

        @Html.ValidationMessageFor(model => model.CHAMADO, "", new { @class = "text-danger"
    })
    </div>
</div>
<div class="form-group">
    @Html.LabelFor(model => model.DATA_CRIACAO_CHAMADO, htmlAttributes: new { @class
= "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.DATA_CRIACAO_CHAMADO, new { htmlAttributes = new
{ @class = "form-control", @readonly = "readonly" } })
        @Html.ValidationMessageFor(model => model.DATA_CRIACAO_CHAMADO, "", new {
@class = "text-danger" })
    </div>
</div>
<div class="form-group">
    @Html.LabelFor(model => model.ID_USUARIOS_RESP, "Usuário", htmlAttributes: new {
@class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("ID_USUARIOS_RESP", null, htmlAttributes: new { @class = "form-
control", @readonly = "readonly" })
        @Html.ValidationMessageFor(model => model.ID_USUARIOS_RESP, "", new { @class =
"text-danger" })
    </div>
</div>
<div class="form-group">
    @Html.LabelFor(model => model.DATA_INICIO_SUPORTE, htmlAttributes: new { @class =
"control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.DATA_INICIO_SUPORTE, new { htmlAttributes = new {
@class = "form-control" } })
        @Html.ValidationMessageFor(model => model.DATA_INICIO_SUPORTE, "", new { @class
= "text-danger" })
    </div>
</div>
<div class="form-group">
    @Html.LabelFor(model => model.SOLUCAO,"Solução", htmlAttributes: new { @class =
"control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.SOLUCAO, new { htmlAttributes = new { @class = "form-
control" } })
        @Html.ValidationMessageFor(model => model.SOLUCAO, "", new { @class = "text-danger"
    })
    </div>
</div>
<div class="form-group">
    @Html.LabelFor(model => model.DATA_FINALIZACAO_CHAMADO, htmlAttributes: new {
@class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.DATA_FINALIZACAO_CHAMADO, new { htmlAttributes =
new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.DATA_FINALIZACAO_CHAMADO, "", new {
@class = "text-danger" })
    </div>
</div>
<div class="form-group">
    @Html.LabelFor(model => model.ID_STATUS_CHAMADO, "Status", htmlAttributes: new {
@class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("ID_STATUS_CHAMADO", null, htmlAttributes: new { @class = "form-
control" })

```

```

        @Html.ValidationMessageFor(model => model.ID_STATUS_CHAMADO, "", new { @class =
"text-danger" })
    </div>
</div>
<div class="form-group">
    @Html.LabelFor(model => model.ID_TECNICO, "Analista", htmlAttributes: new { @class =
"control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("ID_TECNICO", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.ID_TECNICO, "", new { @class = "text-
danger" })
    </div>
</div>
<div class="form-group">
    @Html.LabelFor(model => model.COMENTARIOS, "Comentários",htmlAttributes: new { @class
= "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.COMENTARIOS, new { htmlAttributes = new { @class =
"form-control" } })
        @Html.ValidationMessageFor(model => model.COMENTARIOS, "", new { @class = "text-
danger" })
    </div>
</div>
<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Finalizar Atendimento" class="btn btn-default btn-f btn-entrada"
/>
    </div>
</div>
</div>
}
<script src="~/Scripts/jquery-1.10.2.min.js"></script>
<script src="~/Scripts/jquery.validate.min.js"></script>
<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>

```

View ChamadoAdminController Details.cshtml

```

@model UniSystemHelpDesk.Models.US_CHAMADOS
@{
    ViewBag.Title = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Detalhes do Chamado</h2>
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
            @Html.ActionLink("Atender Chamado", "Edit", new { id = Model.ID_CHAMADO }, new { @class
= "navbar-brand btn-new" })
        </div>
        <div>
            @Html.ActionLink("Voltar", "Index", "ChamadoAdmin", new { area = "" }, new { @class =
"navbar-brand btn-newvs" })
        </div>
    </div>
</div>
<div>
    <hr />
    <dl class="dl-horizontal form-alin2">
        <dt>
            @Html.DisplayName("CÓD CHAMADO")

```

```

</dt>
<dd>
    @Html.DisplayFor(model => model.ID_CHAMADO)
</dd>
<dt>
    @Html.DisplayName("DESCRIÇÃO CHAMADO")
</dt>
<dd>
    @Html.DisplayFor(model => model.CHAMADO)
</dd>
<dt>
    @Html.DisplayName("DATA DE CRIAÇÃO")
</dt>
<dd>
    @Html.DisplayFor(model => model.DATA_CRIACAO_CHAMADO)
</dd>
<dt>
    @Html.DisplayName("DATA DE SUPORTE")
</dt>
<dd>
    @Html.DisplayFor(model => model.DATA_INICIO_SUPORTE)
</dd>
<dt>
    @Html.DisplayName("DATA DE FINALIZAÇÃO")
</dt>
<dd>
    @Html.DisplayFor(model => model.DATA_FINALIZACAO_CHAMADO)
</dd>
<dt>
    @Html.DisplayName("SOLUÇÃO DO CHAMADO")
</dt>
<dd>
    @Html.DisplayFor(model => model.SOLUCAO)
</dd>
<dt>
    @Html.DisplayName("COMENTÁRIOS")
</dt>
<dd>
    @Html.DisplayFor(model => model.COMENTARIOS)
</dd>
<dt>
    @Html.DisplayName("EQUIPAMENTO")
</dt>
<dd>
    @Html.DisplayFor(model => model.US_EQUIPAMENTO.EQUIPAMENTO)
</dd>
<dt>
    @Html.DisplayName("STATUS")
</dt>
<dd>
    @Html.DisplayFor(model => model.US_STATUS.STATUS_CHAMADO)
</dd>
<dt>
    @Html.DisplayName("ANALISTA")
</dt>
<dd>
    @Html.DisplayFor(model => model.US_TECNICO.TECNICO)
</dd>
<dt>
    @Html.DisplayName("USUÁRIO")

```

```

        </dt>
        <dd>
            @Html.DisplayFor(model => model.US_USUARIOS.NOME_USUARIO)
        </dd>
    </dl>
</div>

```

View ChamadoAdminController Delete.cshtml

```

model UniSystemHelpDesk.Models.US_CHAMADOS
@{
    ViewBag.Title = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Excluir Chamado</h2>
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">

            </div>
            <div>
                @Html.ActionLink("Voltar", "Index", "ChamadoAdmin", new { area = "" }, new { @class =
"navbar-brand btn-newvs" })
            </div>
        </div>
    </div>
<hr />
<h3>Tem certeza que deseja excluir esse chamado?</h3>
<div>
    <hr />
    <dl class="dl-horizontal form-alin2">
        <dt>
            @Html.DisplayName("Cód chamado:")
        </dt>
        <dd>
            @Html.DisplayFor(model => model.ID_CHAMADO)
        </dd>
        <dt>
            @Html.DisplayName("Descrição Chamado:")
        </dt>
        <dd>
            @Html.DisplayFor(model => model.CHAMADO)
        </dd>
        <dt>
            @Html.DisplayName("Data de Criação:")
        </dt>
        <dd>
            @Html.DisplayFor(model => model.DATA_CRIACAO_CHAMADO)
        </dd>
        <dt>
            @Html.DisplayName("Data de Inicio Suporte:")
        </dt>
        <dd>
            @Html.DisplayFor(model => model.DATA_INICIO_SUORTE)
        </dd>
        <dt>
            @Html.DisplayName("Data de Finalização:")
        </dt>
        <dd>
            @Html.DisplayFor(model => model.DATA_FINALIZACAO_CHAMADO)
        </dd>
    </dl>

```



```

</dd>
<dt>
    @Html.DisplayName("Solução Chamado:")
</dt>
<dd>
    @Html.DisplayFor(model => model.SOLUCAO)
</dd>
<dt>
    @Html.DisplayName("Comentários:")
</dt>
<dd>
    @Html.DisplayFor(model => model.COMENTARIOS)
</dd>
<dt>
    @Html.DisplayName("Equipamento:")
</dt>
<dd>
    @Html.DisplayFor(model => model.US_EQUIPAMENTO.EQUIPAMENTO)
</dd>
<dt>
    @Html.DisplayName("Status:")
</dt>
<dd>
    @Html.DisplayFor(model => model.US_STATUS.STATUS_CHAMADO)
</dd>
<dt>
    @Html.DisplayName("Atendente:")
</dt>
<dd>
    @Html.DisplayFor(model => model.US_TECNICO.TECNICO)
</dd>
<dt>
    @Html.DisplayName("Usuário:")
</dt>
<dd>
    @Html.DisplayFor(model => model.US_USUARIOS.NOME_USUARIO)
</dd>
</dl>
@using (Html.BeginForm()) {
    @Html.AntiForgeryToken()

    <div class="form-actions no-color">
        <input type="submit" value="Excluir" class="btn btn-default btn-d" />
    </div>
}
</div>

```

UsuarioController

```

namespace UniSystemHelpDesk.Controllers
{
    [Authorize(Roles="Usuário")]
    public class UsuarioController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }
    }
}

```

View UsuarioController Index.cshtml

```
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_LayoutUsuario.cshtml";
}
<h2 style="text-align:center;">Bem vindo Usuario</h2>
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
            @Html.ActionLink("Novo Chamado", "Index", "ChamadoUser/Create", new { area = "" }, new {
                @class = "navbar-brand btn-new" })
            @Html.ActionLink("Meus Chamados", "Index", "ChamadoUser/Index", new { area = "" }, new {
                @class = "navbar-brand btn-new" })
        </div>
        <div>
            @Html.ActionLink("Sair", "Logout", "Conta", new { area = "" }, new { @class = "navbar-brand
                btn-newws" })
        </div>
    </div>
</div>
<hr/>
<div class="alilogo">
    <div class="logop">
        
    </div>
</div>
```

ChamadoUserController

```
namespace UniSystemHelpDesk.Controllers
{
    [Authorize(Roles = "Usuário")]
    public class ChamadoUserController : Controller
    {
        private BDUniSystemHelpDesk db = new BDUniSystemHelpDesk();

        // GET: ChamadoUser
        public ActionResult Index()
        {
            var uS_CHAMADOS = db.US_CHAMADOS.Include(u => u.US_EQUIPAMENTO).Include(u =>
                u.US_STATUS).Include(u => u.US_TECNICO).Where(a => a.US_USUARIOS.EMAIL_USUARIO ==
                User.Identity.Name);
            return View(uS_CHAMADOS.ToList());
        }

        // GET: ChamadoUser/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            US_CHAMADOS uS_CHAMADOS = db.US_CHAMADOS.Find(id);
            if (uS_CHAMADOS == null)
            {
                return HttpNotFound();
            }
        }
    }
}
```

```

        return View(uS_CHAMADOS);
    }

    // GET: ChamadoUser/Create
    public ActionResult Create()
    {
        ViewBag.ID_EQUIPAMENTO = db.US_EQUIPAMENTO.ToList();
        ViewBag.ID_STATUS_CHAMADO = new SelectList(db.US_STATUS.Where(a =>
a.STATUS_CHAMADO == "Aberto"), "ID_STATUS_CHAMADO", "STATUS_CHAMADO");
        ViewBag.ID_TECNICO = new SelectList(db.US_TECNICO, "ID_TECNICO", "TECNICO");
        ViewBag.ID_USUARIOS_RESP = new SelectList(db.US_USUARIOS.Where(a =>
a.EMAIL_USUARIO == User.Identity.Name), "ID_USUARIOS", "NOME_USUARIO");
        return View();
    }

    // POST: ChamadoUser/Create
    // To protect from overposting attacks, please enable the specific properties you want to bind to,
    for
    // more details see http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Create([Bind(Include
=
"ID_CHAMADO,CHAMADO,DATA_CRIACAO_CHAMADO,DATA_INICIO_SUPOORTE,DATA_FINALIZ
ACAO_CHAMADO,SOLUCAO,COMENTARIOS,ID_USUARIOS_RESP,ID_EQUIPAMENTO,ID_STAT
US_CHAMADO,ID_TECNICO")] US_CHAMADOS uS_CHAMADOS)
    {
        if (ModelState.IsValid)
        {
            db.US_CHAMADOS.Add(uS_CHAMADOS);
            db.SaveChanges();
            return RedirectToAction("Index");
        }

        ViewBag.ID_EQUIPAMENTO = new SelectList(db.US_EQUIPAMENTO,
"ID_EQUIPAMENTO", "EQUIPAMENTO", uS_CHAMADOS.ID_EQUIPAMENTO);
        ViewBag.ID_STATUS_CHAMADO = new SelectList(db.US_STATUS,
"ID_STATUS_CHAMADO", "STATUS_CHAMADO", uS_CHAMADOS.ID_STATUS_CHAMADO);
        ViewBag.ID_TECNICO = new SelectList(db.US_TECNICO, "ID_TECNICO", "TECNICO",
uS_CHAMADOS.ID_TECNICO);
        ViewBag.ID_USUARIOS_RESP = new SelectList(db.US_USUARIOS, "ID_USUARIOS",
"NOME_USUARIO", uS_CHAMADOS.ID_USUARIOS_RESP);
        return View(uS_CHAMADOS);
    }

    // GET: ChamadoUser/Edit/5
    public ActionResult Edit(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        US_CHAMADOS uS_CHAMADOS = db.US_CHAMADOS.Find(id);
        if (uS_CHAMADOS == null)
        {
            return HttpNotFound();
        }
        ViewBag.ID_EQUIPAMENTO = new SelectList(db.US_EQUIPAMENTO,
"ID_EQUIPAMENTO", "EQUIPAMENTO", uS_CHAMADOS.ID_EQUIPAMENTO);
        ViewBag.ID_STATUS_CHAMADO = new SelectList(db.US_STATUS,
"ID_STATUS_CHAMADO", "STATUS_CHAMADO", uS_CHAMADOS.ID_STATUS_CHAMADO);

```

```

        ViewBag.ID_TECNICO = new SelectList(db.US_TECNICO, "ID_TECNICO", "TECNICO",
uS_CHAMADOS.ID_TECNICO);
        ViewBag.ID_USUARIOS_RESP = new SelectList(db.US_USUARIOS, "ID_USUARIOS",
"NOME_USUARIO", uS_CHAMADOS.ID_USUARIOS_RESP);
        return View(uS_CHAMADOS);
    }

    // POST: ChamadoUser/Edit/5
    // To protect from overposting attacks, please enable the specific properties you want to bind to,
for
    // more details see http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Edit([Bind(Include
=
"ID_CHAMADO,CHAMADO,DATA_CRIACAO_CHAMADO,DATA_INICIO_SUPOORTE,DATA_FINALIZ
ACAO_CHAMADO,SOLUCAO,COMENTARIOS,ID_USUARIOS_RESP,ID_EQUIPAMENTO,ID_STAT
US_CHAMADO,ID_TECNICO")] US_CHAMADOS uS_CHAMADOS)
    {
        if (ModelState.IsValid)
        {
            db.Entry(uS_CHAMADOS).State = EntityState.Modified;
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        ViewBag.ID_EQUIPAMENTO = new SelectList(db.US_EQUIPAMENTO,
"ID_EQUIPAMENTO", "EQUIPAMENTO", uS_CHAMADOS.ID_EQUIPAMENTO);
        ViewBag.ID_STATUS_CHAMADO = new SelectList(db.US_STATUS,
"ID_STATUS_CHAMADO", "STATUS_CHAMADO", uS_CHAMADOS.ID_STATUS_CHAMADO);
        ViewBag.ID_TECNICO = new SelectList(db.US_TECNICO, "ID_TECNICO", "TECNICO",
uS_CHAMADOS.ID_TECNICO);
        ViewBag.ID_USUARIOS_RESP = new SelectList(db.US_USUARIOS, "ID_USUARIOS",
"NOME_USUARIO", uS_CHAMADOS.ID_USUARIOS_RESP);
        return View(uS_CHAMADOS);
    }

    // GET: ChamadoUser/Delete/5
    public ActionResult Delete(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        US_CHAMADOS uS_CHAMADOS = db.US_CHAMADOS.Find(id);
        if (uS_CHAMADOS == null)
        {
            return HttpNotFound();
        }
        return View(uS_CHAMADOS);
    }

    // POST: ChamadoUser/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public ActionResult DeleteConfirmed(int id)
    {
        US_CHAMADOS uS_CHAMADOS = db.US_CHAMADOS.Find(id);
        db.US_CHAMADOS.Remove(uS_CHAMADOS);
        db.SaveChanges();
        return RedirectToAction("Index");
    }
}

```

```

        protected override void Dispose(bool disposing)
        {
            if (disposing)
            {
                db.Dispose();
            }
            base.Dispose(disposing);
        }
    }
}

```

View ChamadoUserController Index.cshtml

```

@model IEnumerable<UniSystemHelpDesk.Models.US_CHAMADOS>
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Meus Chamados</h2>
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
            @Html.ActionLink("Novo Chamado", "Index", "ChamadoUser/Create", new { area = "" }, new {
                @class = "navbar-brand btn-new" })
        </div>
        <div>
            @Html.ActionLink("Voltar", "Index", "Usuario/Index", new { area = "" }, new { @class = "navbar-
                brand btn-newvs" })
        </div>
    </div>
</div>
<hr />
<table class="table table-l">
    <tr>
        <th>
            @Html.DisplayName("Cód chamado")
        </th>
        <th>
            @Html.DisplayName("Descrição do chamado")
        </th>
        <th>
            @Html.DisplayName("Data de abertura")
        </th>
        <th>
            @Html.DisplayName("Equipamento")
        </th>
        <th>
            @Html.DisplayName("Status")
        </th>
        <th>
            Ações
        </th>
    </tr>
    @foreach (var item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.ID_CHAMADO)
            </td>

```

```

        <td>
            @Html.DisplayFor(modelItem => item.CHAMADO)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.DATA_CRIACAO_CHAMADO)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.US_EQUIPAMENTO.EQUIPAMENTO)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.US_STATUS.STATUS_CHAMADO)
        </td>
        <td>
            <a href="@Url.Action("Edit", "ChamadoUser", new{ id = item.ID_CHAMADO, @title="Editar"
            }}" class="fas fa-edit icons-espa link-hover" title="Editar"></a>
            <a href="@Url.Action("Details", "ChamadoUser", new { id = item.ID_CHAMADO, @title =
            "Detalhes" }}" class="fas fa-search-plus icons-espa link-hover" title="Detalhes"></a>
        </td>
    </tr>
}
</table>

```

View ChamadoUserController Edit.cshtml

```

@model UniSystemHelpDesk.Models.US_CHAMADOS
@{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Editar Chamado</h2>
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
        </div>
        <div>
            @Html.ActionLink("Voltar", "Index", "ChamadoUser", new { area = "" }, new { @class = "navbar-
brand btn-newvs" })
        </div>
    </div>
</div>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <hr />
    <div class="form-horizontal form-alin">
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.ID_CHAMADO)
        <div class="form-group">
            @Html.LabelFor(model => model.ID_EQUIPAMENTO, "Equipamento", htmlAttributes: new {
@class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("ID_EQUIPAMENTO", null, htmlAttributes: new { @class = "form-
control" })
                @Html.ValidationMessageFor(model => model.ID_EQUIPAMENTO, "", new { @class = "text-
danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.CHAMADO, "Descrição do Chamado", htmlAttributes: new {
@class = "control-label col-md-2" })

```

```

        <div class="col-md-10">
            @Html.EditorFor(model => model.CHAMADO, new { htmlAttributes = new { @class = "form-
control" } })
            @Html.ValidationMessageFor(model => model.CHAMADO, "", new { @class = "text-danger"
        })
        </div>
    </div>
    <div class="form-group">
        @Html.LabelFor(model => model.COMENTARIOS, "Comentários", htmlAttributes: new {
@class = "control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.COMENTARIOS, new { htmlAttributes = new { @class =
"form-control" } })
            @Html.ValidationMessageFor(model => model.COMENTARIOS, "", new { @class = "text-
danger" })
        </div>
    </div>
    <div class="form-group">
        @Html.LabelFor(model => model.DATA_CRIACAO_CHAMADO, "Data de abertura",
htmlAttributes: new { @style = "display:none", @class = "control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.DATA_CRIACAO_CHAMADO, new { htmlAttributes = new
{ @style = "display:none", @class = "form-control" } })
            @Html.ValidationMessageFor(model => model.DATA_CRIACAO_CHAMADO, "", new {
@class = "text-danger" })
        </div>
    </div>
    <div class="form-group">
        @Html.LabelFor(model => model.ID_USUARIOS_RESP, "Usuário", htmlAttributes: new {
@style = "display:none", @class = "control-label col-md-2" })
        <div class="col-md-10">
            @Html.DropDownList("ID_USUARIOS_RESP", null, htmlAttributes: new { @style =
"display:none", @class = "form-control" })
            @Html.ValidationMessageFor(model => model.ID_USUARIOS_RESP, "", new { @class =
"text-danger" })
        </div>
    </div>
    <div class="form-group">
        @Html.LabelFor(model => model.ID_STATUS_CHAMADO, "Status ", htmlAttributes: new {
@style = "display:none", @class = "control-label col-md-2" })
        <div class="col-md-10">
            @Html.DropDownList("ID_STATUS_CHAMADO", null, htmlAttributes: new { @style =
"display:none", @class = "form-control" })
            @Html.ValidationMessageFor(model => model.ID_STATUS_CHAMADO, "", new { @class =
"text-danger" })
        </div>
    </div>
    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value="Editar" class="btn btn-default btn-l btn-entrada" />
        </div>
    </div>
</div>
}
<script src="~/Scripts/jquery-1.10.2.min.js"></script>
<script src="~/Scripts/jquery.validate.min.js"></script>
<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>

```

View ChamadoUserController Details.cshtml

```
@model UniSystemHelpDesk.Models.US_CHAMADOS
@{
    ViewBag.Title = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Detalhes do Chamado</h2>
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
            <div>
                @Html.ActionLink("Voltar", "Index", "ChamadoUser", new { area = "" }, new { @class = "navbar-brand btn-newvs" })
            </div>
        </div>
    </div>
</div>
<div>
    <hr />
    <dl class="dl-horizontal form-alin2">
        <dt>
            @Html.DisplayName("CÓD CHAMADO")
        </dt>
        <dd>
            @Html.DisplayFor(model => model.ID_CHAMADO)
        </dd>
        <dt>
            @Html.DisplayName("DESCRIÇÃO CHAMADO")
        </dt>
        <dd>
            @Html.DisplayFor(model => model.CHAMADO)
        </dd>
        <dt>
            @Html.DisplayName("DATA DE ABERTURA")
        </dt>
        <dd>
            @Html.DisplayFor(model => model.DATA_CRIACAO_CHAMADO)
        </dd>
        <dt>
            @Html.DisplayName("DATA DE SUPORTE")
        </dt>
        <dd>
            @Html.DisplayFor(model => model.DATA_INICIO_SUPORTE)
        </dd>
        <dt>
            @Html.DisplayName("DATA DE FINALIZAÇÃO")
        </dt>
        <dd>
            @Html.DisplayFor(model => model.DATA_FINALIZACAO_CHAMADO)
        </dd>
        <dt>
            @Html.DisplayName("SOLUÇÃO")
        </dt>
        <dd>
            @Html.DisplayFor(model => model.SOLUCAO)
        </dd>
        <dt>
            @Html.DisplayName("COMENTÁRIOS")
        </dt>
    </dl>
</div>
```



```

<dd>
    @Html.DisplayFor(model => model.COMENTARIOS)
</dd>
<dt>
    @Html.DisplayName("EQUIPAMENTO")
</dt>
<dd>
    @Html.DisplayFor(model => model.US_EQUIPAMENTO.EQUIPAMENTO)
</dd>
<dt>
    @Html.DisplayName("STATUS")
</dt>
<dd>
    @Html.DisplayFor(model => model.US_STATUS.STATUS_CHAMADO)
</dd>
<dt>
    @Html.DisplayName("ANALISTA")
</dt>
<dd>
    @Html.DisplayFor(model => model.US_TECNICO.TECNICO)
</dd>
<dt>
    @Html.DisplayName("USUÁRIO")
</dt>
<dd>
    @Html.DisplayFor(model => model.US_USUARIOS.NOME_USUARIO)
</dd>
</dl>
</div>

```

View ChamadoUserController Create.cshtml

```

@model UniSystemHelpDesk.Models.US_CHAMADOS
@{
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Novo Chamado</h2>

<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
            @Html.ActionLink("Meus Chamados", "Index", "ChamadoUser/Index", new { area = "" }, new {
                @class = "navbar-brand btn-new" })
        </div>
        <div>
            @Html.ActionLink("Voltar", "Index", "Usuario/Index", new { area = "" }, new { @class = "navbar-
                brand btn-newvs" })
        </div>
    </div>
</div>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <hr />
    <div class="form-horizontal form-alin">
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })

        <div class="form-group">

```

```

        @Html.LabelFor(model => model.ID_EQUIPAMENTO, "Equipamento:", htmlAttributes: new {
@class = "control-label col-md-2" })
        <div class="col-md-10">
            @Html.DropDownList("ID_EQUIPAMENTO", new SelectList(ViewBag.ID_EQUIPAMENTO,
"ID_EQUIPAMENTO", "EQUIPAMENTO"), "Selecione o Equipamento", htmlAttributes: new { @class =
"form-control" })
            @Html.ValidationMessageFor(model => model.ID_EQUIPAMENTO, "", new { @class = "text-
danger" })
        </div>
    </div>
    <div class="form-group">
        @Html.LabelFor(model => model.CHAMADO, "Descrição:", htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.CHAMADO, new { htmlAttributes = new { @class = "form-
control" } })
            @Html.ValidationMessageFor(model => model.CHAMADO, "", new { @class = "text-danger"
})
        </div>
    </div>
    <div class="form-group">
        @Html.LabelFor(model => model.DATA_CRIACAO_CHAMADO, htmlAttributes: new { @class
= "control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.DATA_CRIACAO_CHAMADO, new { htmlAttributes = new
{ @class = "form-control" } })
            @Html.ValidationMessageFor(model => model.DATA_CRIACAO_CHAMADO, "", new {
@class = "text-danger" })
        </div>
    </div>
    <div class="form-group">
        @Html.LabelFor(model => model.ID_USUARIOS_RESP, "ID_USUARIOS_RESP",
htmlAttributes: new { @style = "display:none", @class = "control-label col-md-2" })
        <div class="col-md-10">
            @Html.DropDownList("ID_USUARIOS_RESP", null, htmlAttributes: new { @style =
"display:none", @class = "form-control" })
            @Html.ValidationMessageFor(model => model.ID_USUARIOS_RESP, "", new { @class =
"text-danger" })
        </div>
    </div>
    <div class="form-group">
        @Html.LabelFor(model => model.ID_STATUS_CHAMADO, "ID_STATUS_CHAMADO",
htmlAttributes: new { @style = "display:none", @class = "control-label col-md-2" })
        <div class="col-md-10">
            @Html.DropDownList("ID_STATUS_CHAMADO", null, htmlAttributes: new { @style =
"display:none", @class = "form-control" })
            @Html.ValidationMessageFor(model => model.ID_STATUS_CHAMADO, "", new { @class =
"text-danger" })
        </div>
    </div>
    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value="Criar" class="btn btn-default btn-l btn-entrada" />
        </div>
    </div>
</div>
}
<script src="~/Scripts/jquery-1.10.2.min.js"></script>
<script src="~/Scripts/jquery.validate.min.js"></script>
<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>

```

ChamadoDAO.cs

```
namespace UniSystemHelpDesk.DAL
{
    public class ChamadoDAO
    {
        BDUUniSystemHelpDesk bd = new BDUUniSystemHelpDesk();

        public List<ConsultaU> Consultar()
        {
            BDUUniSystemHelpDesk bd = new BDUUniSystemHelpDesk();
            var novo = bd.ConsultaUS.ToList();
            return novo;
        }
    }
}
```

WebServiceUS.asmx.cs

```
namespace UniSystemHelpDesk.WebService
{
    /// <summary>
    /// Summary description for WebServiceUS
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following
    line.
    // [System.Web.Script.Services.ScriptService]
    public class WebServiceUS : System.Web.Services.WebService
    {
        [WebMethod]
        public List<ConsultaU> ConsultarChamado()
        {
            ChamadoDAO chamadodao = new ChamadoDAO();
            return chamadodao.Consultar();
        }
    }
}
```

APÊNCIDE B – Código WPF .NET

frmPrincipal.xaml

```
<Window x:Class="UniSystemHelpDeskWPF.Apresentacao.frmPrincipal"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="frmPrincipal" Height="643" Width="1277" >
    <Grid>
        <Menu HorizontalAlignment="Left" Height="100" Margin="10,10,0,0" VerticalAlignment="Top"
Width="1249">
            <MenuItem Header="Consulta" Width="350" Height="40" Click="MenuItem_Click"
FontSize="25">
                <MenuItem.Icon>
                    <Image Source="/IMG/serachicon.png"/></Image>
                </MenuItem.Icon>
                <MenuItem Header="Consulta Chamado" HorizontalAlignment="Left" Width="350"
Height="40"/>
            </MenuItem>
        </Menu>
        <Image Source="/IMG/LogoUniSystemNEW.png" Margin="0,58,0,-58" ></Image>
    </Grid>
</Window>
```

frmPrincipal.xaml.cs

```
namespace UniSystemHelpDeskWPF.Apresentacao
{
    /// <summary>
    /// Interaction logic for frmPrincipal.xaml
    /// </summary>
    public partial class frmPrincipal : Window
    {
        public frmPrincipal()
        {
            InitializeComponent();
        }

        private void MenuItem_Click(object sender, RoutedEventArgs e)
        {
            MainWindow frmC = new MainWindow();
            frmC.ShowDialog();
        }
    }
}
```

frmConsulta.xaml

```
<Window x:Name="frmPrincipal" x:Class="UniSystemHelpDeskWPF.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="Consulta de chamados" Height="642.537" Width="1276.865"
Loaded="frmPrincipal_Loaded">
    <Window.Resources>
        <Style TargetType="Button" x:Key="TabButton">
            <Setter Property="Background" Value="White" />
            <Setter Property="TextBlock.TextAlignment" Value="Center" />
            <Setter Property="Template">
                <Setter.Value>
```

```

        <ControlTemplate TargetType="Button">
            <Border    CornerRadius="4"    Background="#00cccc"    BorderBrush="#ccc"
BorderThickness="0,1,1,0" TextBlock.Foreground="AntiqueWhite">
                <ContentPresenter x:Name="contentPresenter" ContentTemplate="{TemplateBinding
ContentTemplate}" Content="{TemplateBinding Content}" HorizontalAlignment="{TemplateBinding
HorizontalContentAlignment}" Margin="{TemplateBinding Padding}"
VerticalAlignment="{TemplateBinding VerticalContentAlignment}"/>
            </Border>
        </ControlTemplate>
    </Setter.Value>
</Setter>
</Style>
</Window.Resources>
<Grid>
    <DataGrid        x:Name="grid"        ColumnWidth="177"        HorizontalAlignment="Left"
AutoGenerateColumns="true"        AutoGeneratingColumn="AutoGeneratingColumn"
SelectionChanged="grid_SelectionChanged" Width="1249" Margin="10,0,0,217">

        </DataGrid>

    <Button    Style="{StaticResource TabButton}"    x:Name="btnAtualizar"    FontSize="35"
Content="Atualizar" HorizontalAlignment="Left" Margin="503,528,0,0" VerticalAlignment="Top"
Width="251"    FontWeight="Bold"    FontFamily="Lucida Sans"    Background="White"
Click="btnAtualizar_Click_1"/>
</Grid>
</Window>

```

frmConsulta.xaml.cs

```

namespace UniSystemHelpDeskWPF
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }

        private void frmPrincipal_Loaded(object sender, RoutedEventArgs e)
        {
            try
            {
                WebServiceUSSoapClient client = new WebServiceUSSoapClient();
                grid.ItemsSource = client.ConsultarChamado();
            }
            catch (Exception)
            {
                MessageBox.Show("Webservice Indisponivel");
            }
        }

        private void grid_SelectionChanged(object sender, SelectionChangedEventArgs e)
        {
        }

        private void AutoGeneratingColumn(object sender, DataGridAutoGeneratingColumnEventArgs e)
        {
            if (e.PropertyName == "ExtensionData")

```

```

    {
        e.Column = null;
    }
    if (e.PropertyName == "DATA_FINALIZACAO_CHAMADO")
        (e.Column as DataGridTextColumn).Binding.StringFormat = "dd/MM/yyyy";
    if (e.PropertyName == "DATA_CRIACAO_CHAMADO")
        (e.Column as DataGridTextColumn).Binding.StringFormat = "dd/MM/yyyy";

    //Alterar Nome das colunas datagridviewer
    if (e.PropertyName.StartsWith("TECNICO"))
    {
        e.Column.Header = "Técnico";
    }
    if (e.PropertyName.StartsWith("STATUS_CHAMADO"))
    {
        e.Column.Header = "Status";
    }
    if (e.PropertyName.StartsWith("DATA_FINALIZACAO"))
    {
        e.Column.Header = "Data finalização";
    }
    if (e.PropertyName.StartsWith("DATA_CRIACAO_CHAMADO"))
    {
        e.Column.Header = "Data abertura";
    }
    if (e.PropertyName.StartsWith("CHAMADO"))
    {
        e.Column.Header = "Descrição de chamado";
    }
    if (e.PropertyName.StartsWith("ID_CHAMADO"))
    {
        e.Column.Header = "Cód chamado";
    }
    if (e.PropertyName.StartsWith("NOME_USUARIO"))
    {
        e.Column.Header = "Usuário";
    }
}

private void btnAtualizar_Click(object sender, RoutedEventArgs e)
{
    frmPrincipal_Loaded(sender, e);
}

private void btnAtualizar_Click_1(object sender, RoutedEventArgs e)
{
    frmPrincipal_Loaded(sender, e);
}
}

```