**UNIVERSIDADE PAULISTA – UNIP**

Jorge Willian dos Santos - D44830-7

Kaique Curan Ribeiro - N1079E-1

Leandro da Silva Oliveira - D32091-2

Matheus Del Poço Nanuk – D4688B-0

Wesley Aparecido Nunes de Assis - D399IB-1

**HELP DESK UNIINFO**

**SOROCABA**

**2018**

Jorge Willian dos Santos - D44830-7

Kaique Curan Ribeiro - N1079E-1

Leandro da Silva Oliveira - D32091-2

Matheus Del Poço Nanuk – D4688B-0

Wesley Aparecido Nunes de Assis - D399IB-1

**HELP DESK UNIINFO**

Trabalho de conclusão de semestre para obtenção do título de graduação em (ADS) apresentado a universidade paulista – UNIP

Orientador: Prof. Richardson Luz

**SOROCABA**

**2018**

CIP - Catalogação na Publicação

Help desk Uniinfo / Jorge Willian dos Santos...[et al.]. - 2018.

0083 f. : il. color

Trabalho de Conclusão de Curso (Graduação) apresentado ao Instituto

de Ciência Exatas e Tecnologia da Universidade Paulista, Sorocaba,

2018.

Área de Concentração: Tecnologia da Informação.

Orientador: Prof. Esp. Richardson Luz.

Coorientadores: Prof. Esp. Eduardo Simi, Prof. Me. Elizeu Fachini,

Prof. Esp. Reverdan Sparinger.

1. Help desk. 2. Engenharia de software. 3. Chamados. 4. Resolução.

I. Santos, Jorge Willian dos. II. Luz, Richardson (orientador). III. Simi,

Eduardo (coorientador). IV. Fachini, Elizeu (coorientador). V. Sparinger,

Reverdan (coorientador).

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

Jorge Willian dos Santos - D44830-7

Kaique Curan Ribeiro - N1079E-1

Leandro da Silva Oliveira - D32091-2

Matheus Del Poço Nanuk – D4688B-0

Wesley Aparecido Nunes de Assis - D399IB-1

**HELP DESK UNIINFO**

Trabalho de conclusão de semestre para obtenção do título de graduação em (ADS) apresentado à universidade paulista – UNIP

APROVADO EM:

BANCA EXAMINADORA

Prof. Reverdan Sparinger

Universidade Paulista - UNIP

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_/\_\_\_

Prof: Wandivaldi Antonio Colla

Universidade Paulista - UNIP

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_/\_\_\_

Prof: Marcelo Tirolli

Universidade Paulista - UNIP

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_/\_\_\_

Prof: Randal Gasparini

Universidade Paulista - UNIP

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_/\_\_\_

*“Todos os homens têm, por*

*natureza, desejo de conhecer”.*

(Aristóteles.)

**RESUMO**

O projeto tratou de um estudo sobre o desenvolvimento de um sistema Help Desk, onde o mesmo foi aplicado em dois modelos: modelo Desktop e modelo Web.

O sistema Help Desk da Uniinfo foi desenvolvido a fim de sanar possíveis complicações referentes ao departamento de TI, avaliando e determinando o grau de cada problema. Estabelecendo, assim, o tempo para que cada vício seja corrigido.

A plataforma Web foi desenvolvida para os Usuários e Administradores do sistema. No projeto, os Administradores têm como função incluir novos tipos de chamados e cadastrar novos usuários, bem como “Editar” e “Excluir” os Dados. Já os Usuários, possuem atribuições diferentes: Abertura de chamados com referência ao problema listado e Descrição do problema.

A plataforma “Desktop” foi desenvolvida especialmente para os Administradores, com a possibilidade de observar todos os chamados, sejam eles em qualquer estado. Atingindo, assim, um controle administrativo total destes.

Sendo assim, o software foi gerado especialmente para a empresa que pretende estruturar seu sistema de “Abertura de Chamados”, o que facilitará aos Usuários a resolução dos problemas do dia a dia de uma forma mais ágil e eficiente. Haja vista que, o sistema permite a administração e o acompanhamento dos mesmos por meio de uma interface Web agradável e objetiva, como também o “Desktop” para o gerenciamento de todas as ações do sistema.

**Palavras-chave:** Help Desk, Chamados, Resolução.

**ABSTRACT**

The project dealt with a study about the designing of a Help Desk system, where it was applied in two models: Desktop model and Web model.

The Uniinfo Help Desk system was designed to address potential complications of the IT department by evaluating and determining the extent of each problem. Establishing, therefore, the time for each ticket to be corrected.

The Web platform was designed for System Users and Administrators. In the project, the Administrators has as function to include new types of tickets and to register new users, as well as "Edit" and "Delete" the Data. Users, however, have different assignments: Ticket opening with reference to the problem listed and Description of the problem.

The "Desktop" platform was specially designed for the Administrators, with the possibility of observing all tickets, in any state that they are. Achieving total control of these.

Therefore, the software was generated especially for the company that intends to structure its "Ticket Opening" system, which will make it easier for Users to solve their day-to-day problems in a more agile and efficient way. Since the system allows the managing and monitoring of all tickets through a nice and objective Web interface, as well as the "Desktop" to manage all the actions of the system.

**Key words:** Help Desk, Called, Resolution.

Sumário

[1 HELP DESK UNIINFO 15](#_Toc531710015)

[2 DESENVOLVIMENTO 17](#_Toc531710016)

[2.1 Tecnologias e Ferramentas utilizadas 17](#_Toc531710017)

[2.1.1 C Sharp 17](#_Toc531710018)

[2.1.2 Framework Entity 17](#_Toc531710019)

[2.1.3 SQL Server 18](#_Toc531710020)

[2.1.4 ASP.NET 18](#_Toc531710021)

[2.1.5 MVC 19](#_Toc531710022)

[2.1.6 WPF 19](#_Toc531710023)

[2.1.7 HTML 20](#_Toc531710024)

[2.1.8 CSS 20](#_Toc531710025)

[2.1.9 Javascrypt 20](#_Toc531710026)

[2.1.10 Bootstrap 20](#_Toc531710027)

[3 HELP DESK PLATAFORMA WEB 21](#_Toc531710028)

[3.1 Imagens das telas da plataforma web 21](#_Toc531710029)

[3.1.1 Imagens das telas da plataforma web como administrador 23](#_Toc531710030)

[3.1.2 Imagens das telas da plataforma web como Usuário 28](#_Toc531710031)

[4 HELP DESK DESKTOP 31](#_Toc531710032)

[4.1 Imagens das telas desktop 31](#_Toc531710033)

[5 WEB SERVICE 33](#_Toc531710034)

[6 BANCO DE DADOS 35](#_Toc531710035)

[6.1 Modelo lógico 36](#_Toc531710036)

[7 DESENVOLVIMENTO DE SOFTWARE ÁGIL COM SCRUM 37](#_Toc531710037)

[7.1 Product Backlog (Backlog do Produto) 37](#_Toc531710038)

[7.1.1 Levantamento dos requisitos. 38](#_Toc531710039)

[7.2 Sprint backlog 38](#_Toc531710040)

[8 UML 39](#_Toc531710041)

[8.1 Diagrama de Caso de Uso 39](#_Toc531710042)

[8.2 Diagrama de Classe 40](#_Toc531710043)

[8.3 Diagrama de sequência 41](#_Toc531710044)

[9 PADRÕES DE QUALIDADE - ISO/IEC 9126 (NBR 13596) 43](#_Toc531710045)

[10 CONCLUSÃO 45](#_Toc531710046)

[REFERÊNCIAS BIBLIOGRÁFICAS 47](#_Toc531710047)

[APÊNDICE 49](#_Toc531710048)

[APÊNDICE A - CÓDIGO FONTE ASP.NET 51](#_Toc531710049)

[APÊNDICE B - CÓDIGO FONTE WPF 161](#_Toc531710050)

# **HELP DESK UNIINFO**

O sistema de Help Desk Uniinfo foi desenvolvido como uma solução para administradores que desejam organizar seu sistema de abertura de chamados. Pensado para organizações de pequeno porte, o sistema permite de forma simples a abertura, administração e acompanhamento de chamados através de uma interface web agradável e limpa

Help desk é um termo em inglês que significa literalmente "balcão de ajuda" e que se refere a um serviço de atendimento aos clientes que procuram por soluções, esclarecimentos sobre dúvidas e outras solicitações para problemas técnicos relacionados a telefonia, informática, tecnologia da informação ou ainda pré e pós-vendas.

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.

Por tratar-se de um sistema de atendimento flexível, o trabalho pode ser realizado tanto pessoalmente como por meio de sistemas de comunicação, como e-mail, telefone, fórum e diferentes outros meios de comunicação.

 Imagem 1 - Logo Uniinfo

Fonte – Uniinfo 2018

# DESENVOLVIMENTO

Nosso sistema foi desenvolvido com as tecnologias e linguagens de programação mais utilizadas no mercado de trabalho atualmente, isso garante aos nossos usuários (administradores e comuns): segurança, fluidez e estabilidade no seu uso diário.

Em seu Back-end está presente o C# por meio do WPF (Windows Prsentation Foundation) e ASP.NET, que possibilitam um maior controle de erros e o desenvolvimento de uma interface de usuário otimizada para uso. Junto ou nosso banco de dados SQL, o Help Desk Uniinfo interage de forma nativa com seus dados e funções de gerenciamento.

Pensado e desenvolvido em uma plataforma WEB, sua interface difere entre funções comuns (Usuários básicos) e funções administrativas (Administradores do sistema, TI):

## Tecnologias e Ferramentas utilizadas

### C Sharp

A linguagem C# foi criada juntamente com a arquitetura da plataforma. NET da Microsoft. É uma junção das principais linguagens, aperfeiçoando as execuções e adicionando novos recursos, tornando-a mais atrativa para os desenvolvedores que queiram migrar para Microsoft. NET.

A linguagem C# é orientada à objetos, suportando conceitos como encapsulamento, herança e polimorfismo. Todas suas variáveis e métodos são encapsulados dentro das definições de uma classe. Sendo usada na maioria das classes do .NET framework. Foi a primeira linguagem orientada a componentes da família C/C++, segue os padrões POO, onde tudo deriva de um ancestral comum. Sendo assim, a linguagem C# é System.Object.

### Framework Entity

O Microsoft Entity Framework é uma ferramenta de mapeamento do objeto relacional (ORM – Object Relational Management), que permite aos desenvolvedores trabalhar com classes (entidades), que correspondem as tabelas em um banco de dados, tornando-a transparentes os acessos a estes dados e principalmente, eliminando a necessidade de escrever código de banco de dados (SELECT, INSERT, UPDATE, DELETE) na aplicação. Com o Entity Framework os desenvolvedores manipulam os dados através de classes que são mapeadas com as tabelas do banco, assim, o acesso e manipulação destes dados fica mais simples, pois os desenvolvedores já trabalham naturalmente com os objetos, propriedades e coleções no seu desenvolvimento.

Os comandos submetidos pelo Entity Framework são “traduzidos” para a linguagem do banco de dados através do seu provider, gerando os comandos SQL mais adequados para cada operação e principalmente, que tenham o máximo de desempenho.

### SQL Server

Criado pela Microsoft, o Software Gerenciador de Banco de Dados (SGBD) foi lançado originalmente em 1988, em uma parceria com a empresa Sybase. Na época, funcionava exclusivamente em equipamentos com o sistema operacional OS/2. A partir de 1994, foi disponibilizada uma versão para Windows, o que ajudou a popularizar e sedimentar o produto no mercado.

O SQL Server é um gerenciador de bancos de dados relacional, o que significa que as informações que manipulam estão armazenadas em campos de tabelas. Para facilitar o entendimento, imagine uma tabela de clientes, em que cada linha contém os registros de nome, endereço e estado civil. São estruturas como essa que o banco controla. Adicionalmente, o sistema oferece recursos avançados para facilitar a atualização dos dados e garantir que as informações armazenadas se mantenham corretas e confiáveis.

### ASP.NET

É um framework para desenvolvimento de aplicações web na plataforma .Net, está integrado do .Net Framework. Podemos utilizar qualquer linguagem da plataforma .Net, ou linguagens homologadas para a plataforma e desenvolver aplicações web. Qualquer linguagem da plataforma .Net acessa os mesmos recursos do .Net Framework, a linguagem mais comum e utilizadas são: Visual Basic.Net e C#. Faz uso do ADO.Net para acesso a dados, bem como para manipulação dos mesmos.

### MVC

Atualmente, muitos softwares e frameworks estão utilizando do padrão MVC para o desenvolvimento de seus aplicativos/sites. O MVC (Model, View e Controller) é uma arquitetura ou padrão que lhe permite dividir as funcionalidades do sistema/site em camadas, essa divisão é realizada para facilitar resolução de um problema maior. Onde possuímos três camadas básicas, e cada uma delas, com suas características e funções bem definidas.

Model é utilizado para manipular informações de forma mais detalhada, sendo recomendado sempre que possível, se utilize dos modelos para realizar consultas, cálculos e todas as regras de negócio. É o modelo que tem acesso a toda e qualquer informação sendo essa vinda de um banco de dados, arquivo XML.

View é responsável por tudo que o usuário final visualiza, toda a interface, informação, não importando sua fonte de origem, é exibida graças a camada de visão.

Controller é responsável por controlar todo o fluxo de informação que passa pelo site/sistema. Define quais informações devem ser geradas, quais regras devem ser acionadas e para onde as informações devem ir, é nela que essas operações devem ser executadas. Em resumo, é a controladora que executa uma regra de negócio (modelo) e repassa a informação para a visualização (visão).

### WPF

O WPF foi projetado para criar sistemas de apresentação dinâmicos, controlados por dados. Cada parte do sistema foi projetada para criar objetos através de conjuntos de propriedades que controlam o comportamento. A associação de dados é uma parte fundamental do sistema e está integrada em todas as camadas. É possível definir propriedades, usar objetos e associar dados praticamente da mesma forma que poderia fazer usando Windows Forms ou ASP.NET. Com uma investigação mais profunda da arquitetura do WPF, descobrimos que existe a possibilidade de criar aplicativos muito mais completos que, tratam os dados como o controlador principal do aplicativo.

### HTML

Html (Hyper Text Markup Language)linguagem de Marcação de Hiper Texto é utilizada normalmente para criação de sites, é uma linguagem de marcação que estrutura, exibe e organiza todas as informações contidas no site, voltada hoje em dia apenas na semântica e distribuição do conteúdo da página utilizando o CSS e JavaScript, é construída a navegação.

### CSS

CSS (Cascading Style Sheets) folha de estilo em Cascata, esta linguagem é responsável por estilizar o site, afim de que torne a sua visualização melhor, com interferência cores, organização, semântica, sua interação com o HTML é de ser requisitado através da chamada do index.html e realçar a interface, e qualidade na visualização do site. As linhas de códigos abaixo resumem o processo de inserção de imagem na página e definição de seu tamanho.

### Javascrypt

Serve normalmente para manipular todas as ações e eventos do seu Web site, como por exemplo, o click em um botão qualquer, ou um link redirecionando para outra página. Tem a função de gerar a interação com o usuário desde movimentos na tela até momento que é requisitado alguma ação do usuário.

### Bootstrap

Foi desenvolvido com o intuito de otimizar o desenvolvimento de sua plataforma através da adoção de uma estrutura única. Isto reduziria inconsistências entre as diversas formas de se codificar, que variam de profissional para profissional, com sua interface bastante amigável, o Bootstrap oferece uma enorme variedade de Plugins e temas compatíveis com o framework. Além disto, possui integração com qualquer linguagem de programação. O Bootstrap torna o desenvolvimento front-end mais rápido e fácil.

# HELP DESK PLATAFORMA WEB

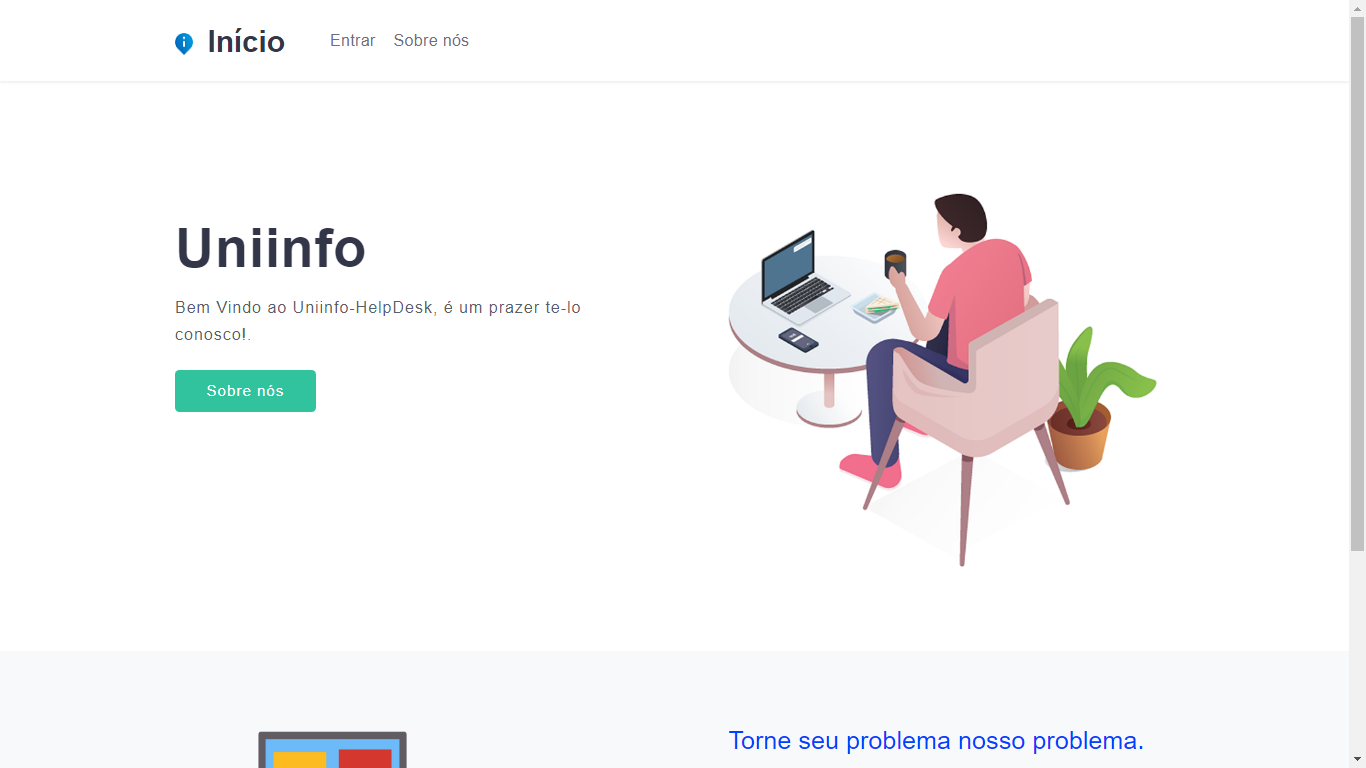
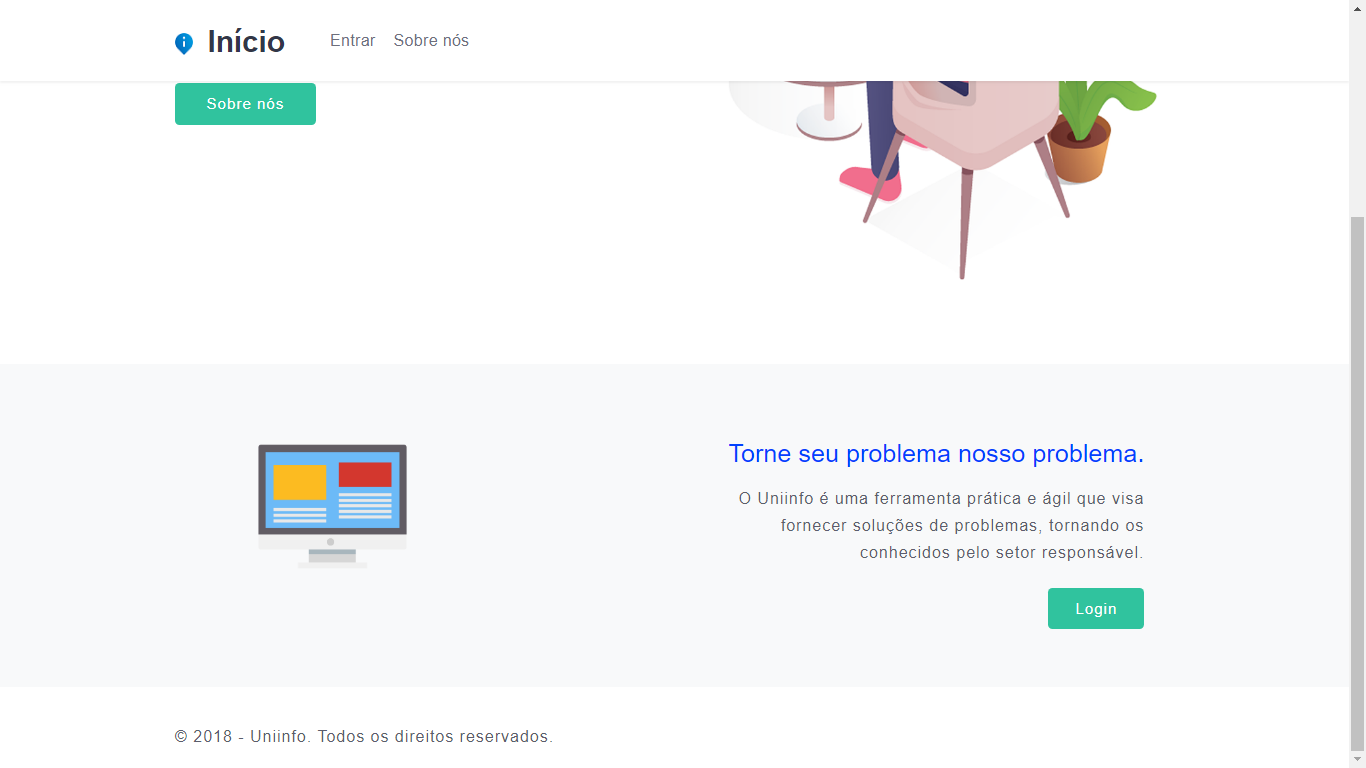
O Help Desk da Uniinfo foi desenvolvido com o objetivo de resolver os chamados (problemas), dos mais variáveis níveis. Os funcionários que estão com problemas abrem um chamado de atendimento e aguardam uma posição de quando o seu problema será avaliado. Para esses acessos criamos os usuários (funcionários e administradores).

O usuário com o acesso de funcionário, tem como controle iniciar um chamado e E-mail para contato, já o usuário com acesso de Login (administrador), tem como controle o cadastro de funcionários, administradores e chamados, fazer a edição de funcionários, administradores e chamados, exclusão de funcionários e chamados, atender um chamado, finalizar um chamado, visualizar um chamado em andamento,

visualizar um chamado e qual funcionário iniciou.

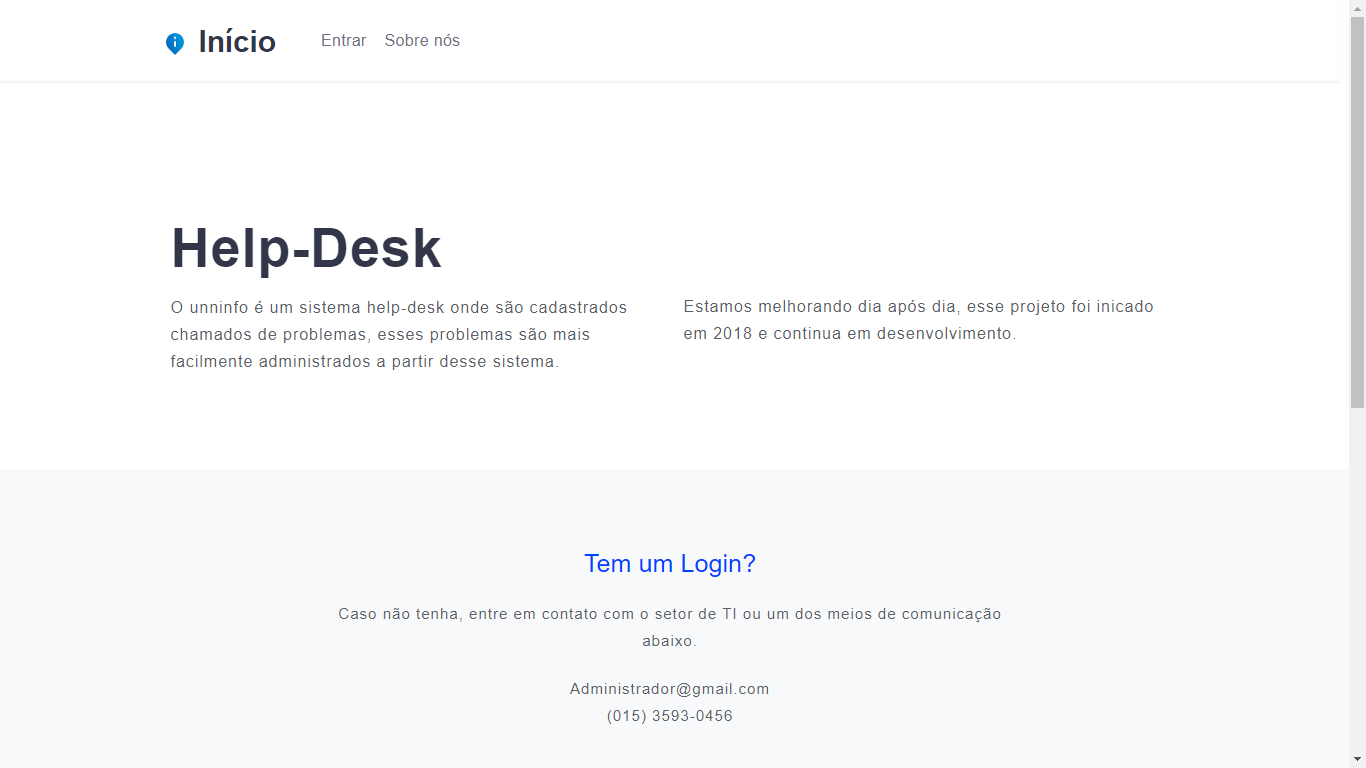
## Imagens das telas da plataforma web

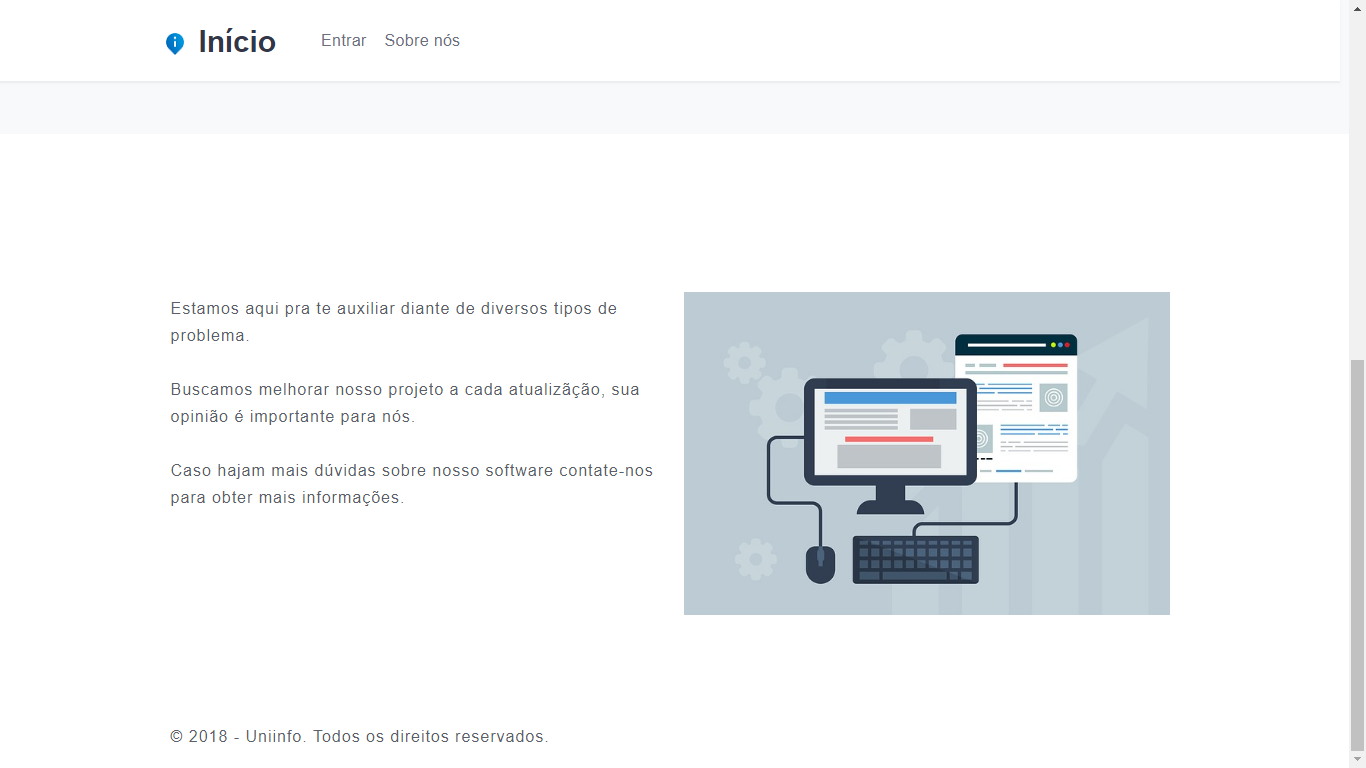
Imagem 2 – Tela inicial Pagina web



Fonte – Uniinfo 2018

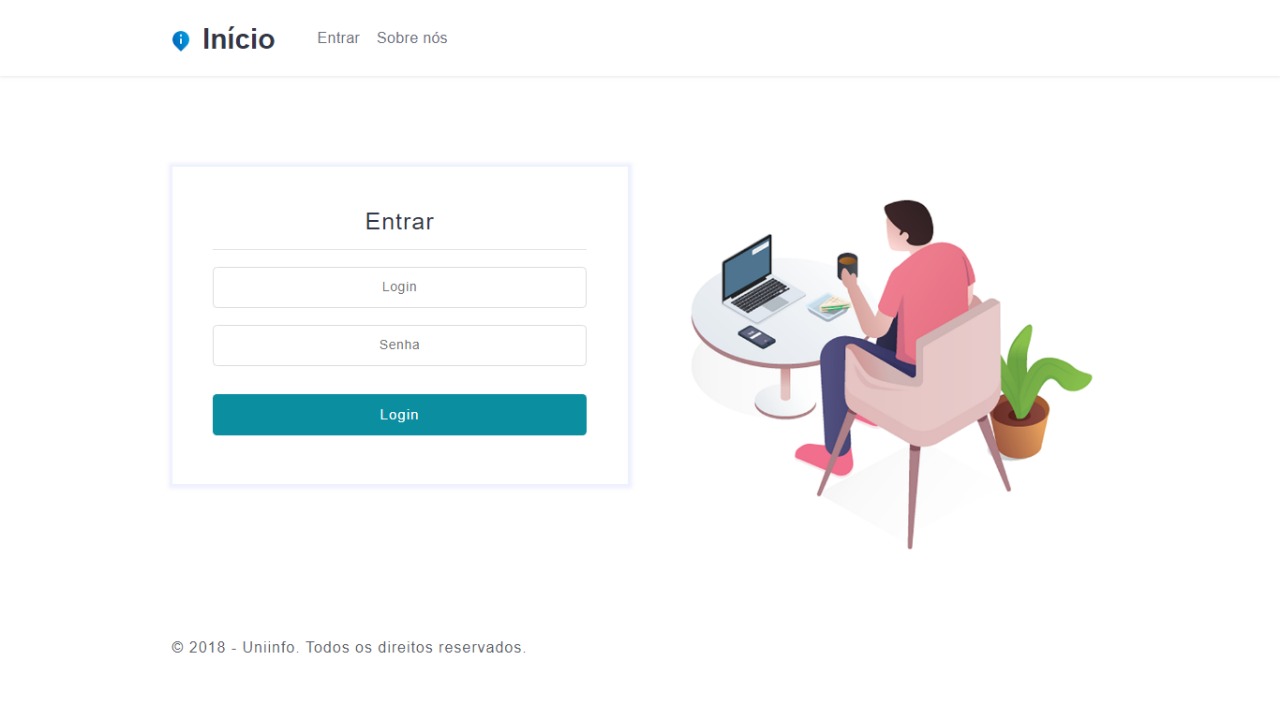
Imagem 3 – Tela Sobre nós Pagina web





Fonte – Uniinfo 2018

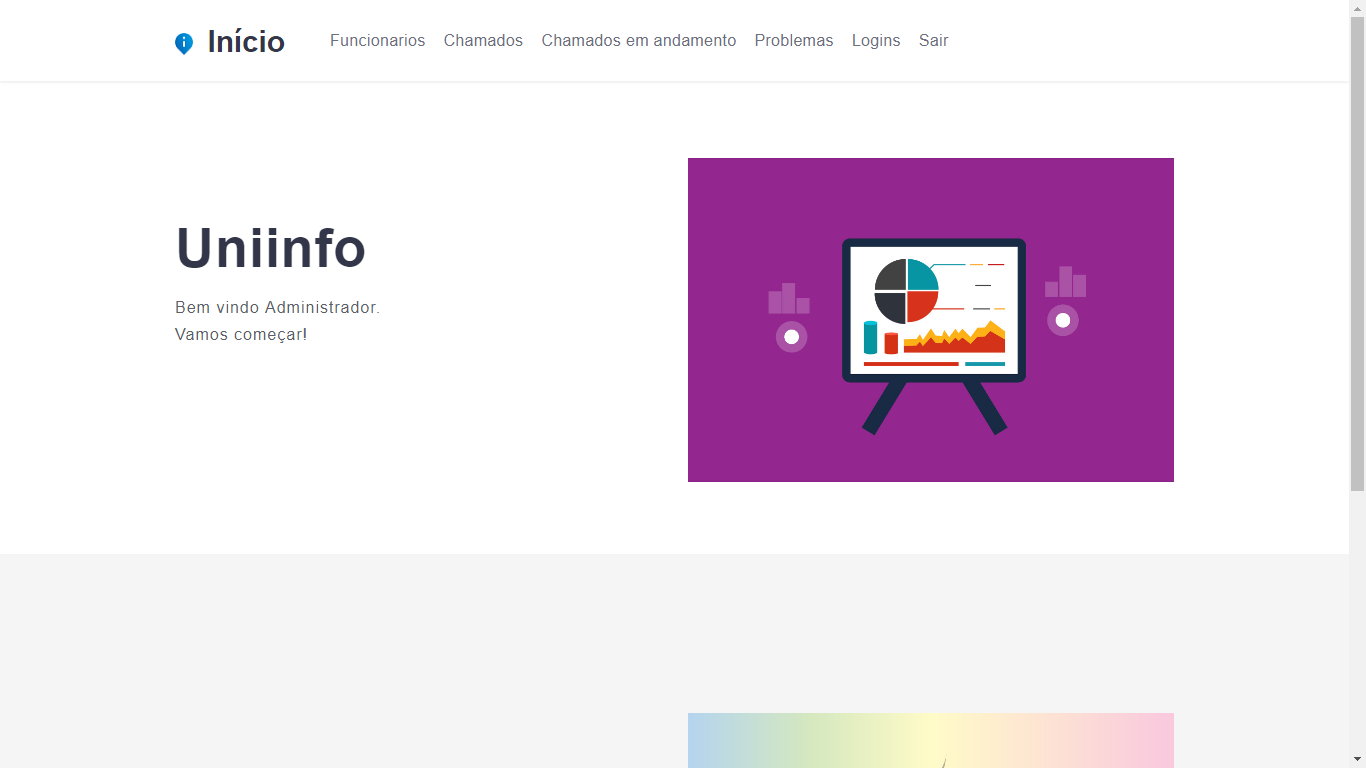
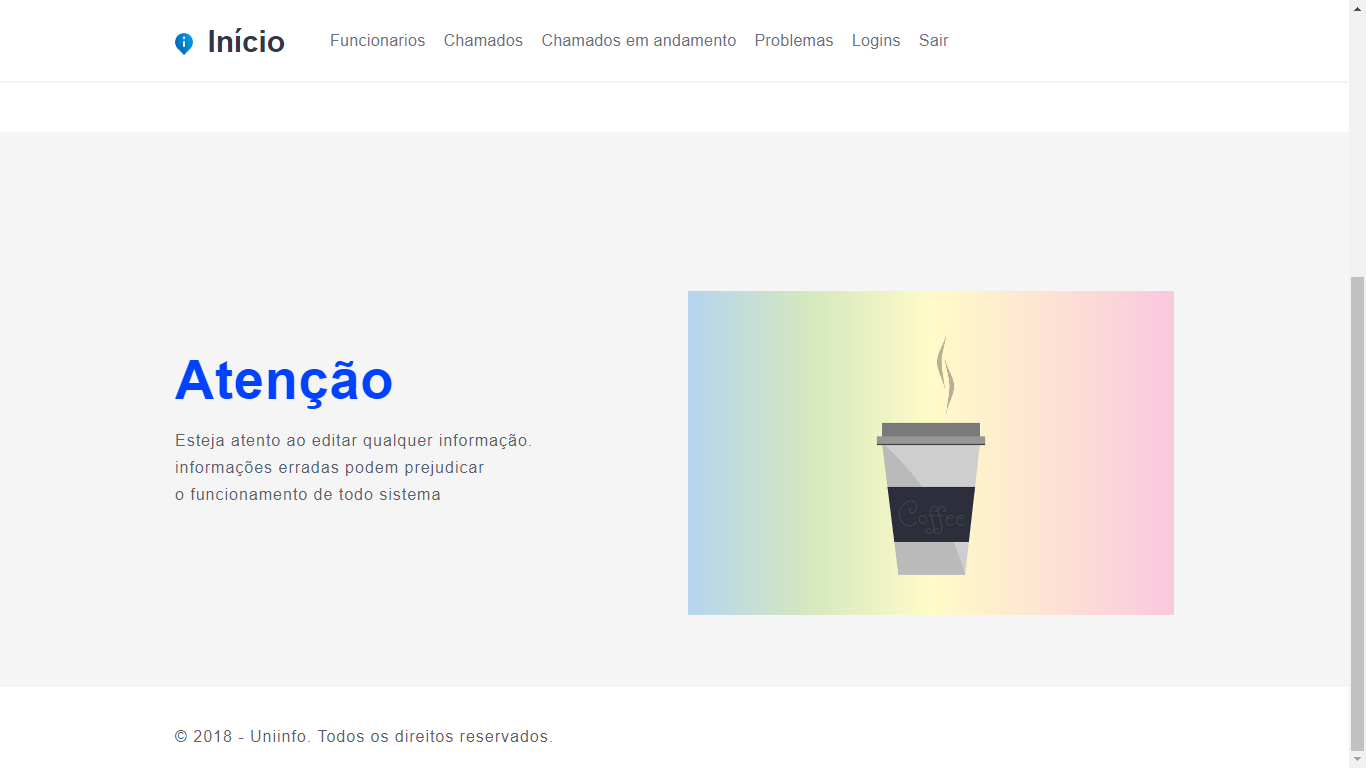
Imagem 4 – Tela Loguin Pagina web



Fonte – Uniinfo 2018

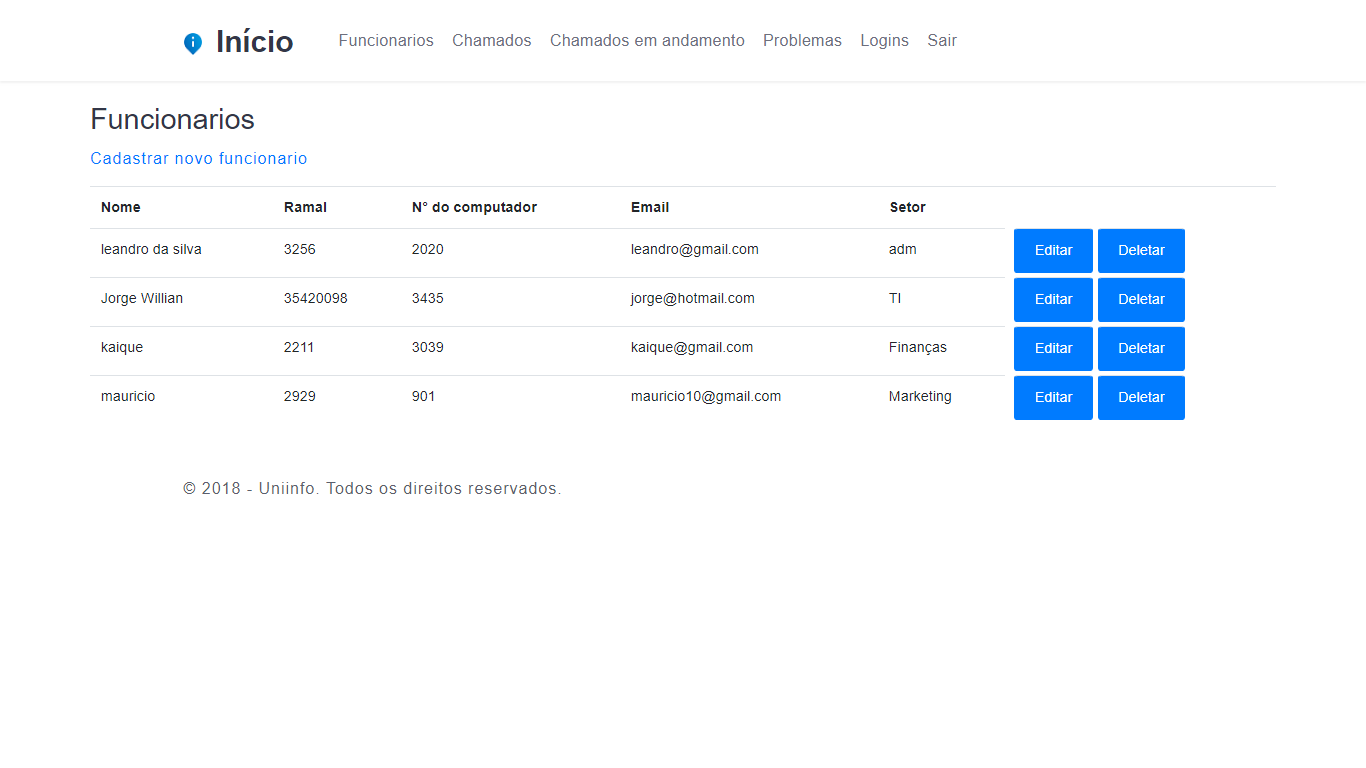
### Imagens das telas da plataforma web como administrador

Imagem 5 – Tela inicial Pagina web administrador



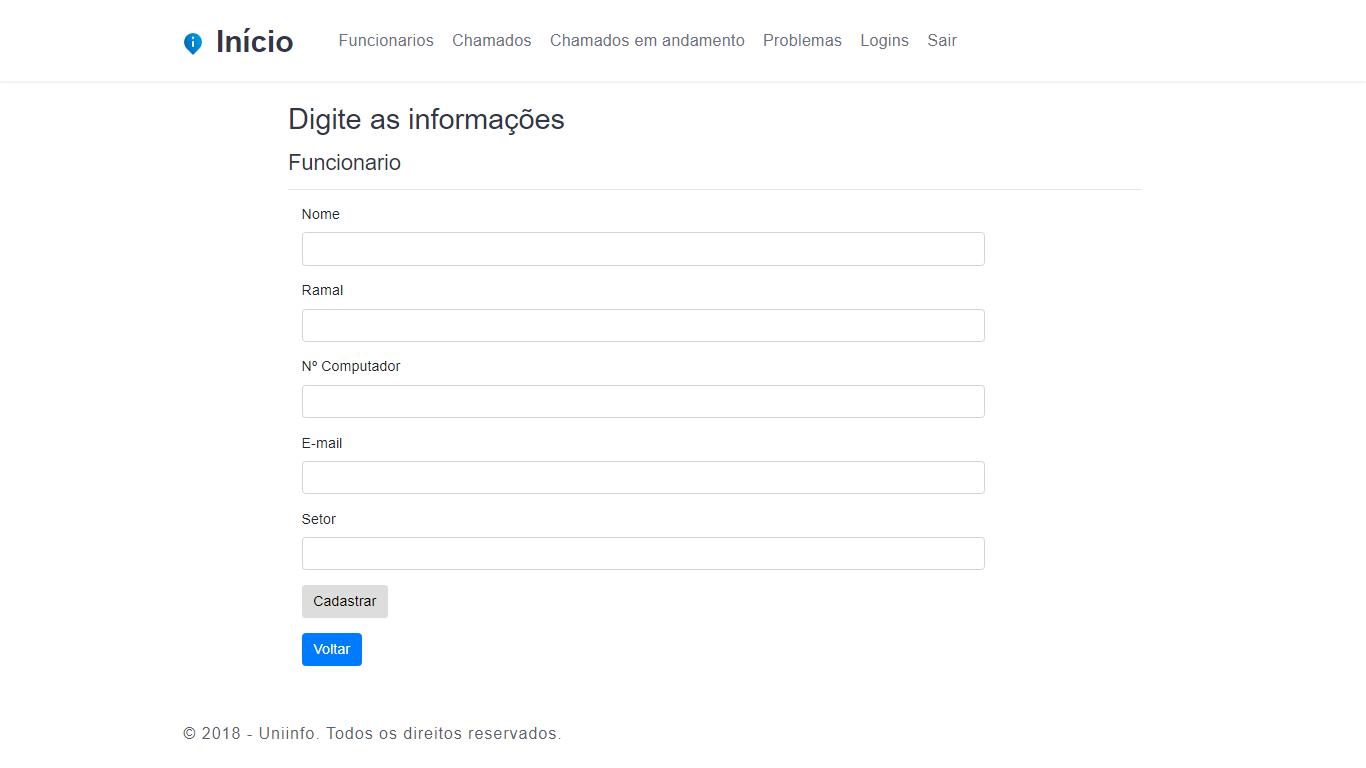
Fonte – Uniinfo 2018

Imagem 6 – Tela funcionário Pagina web administrador



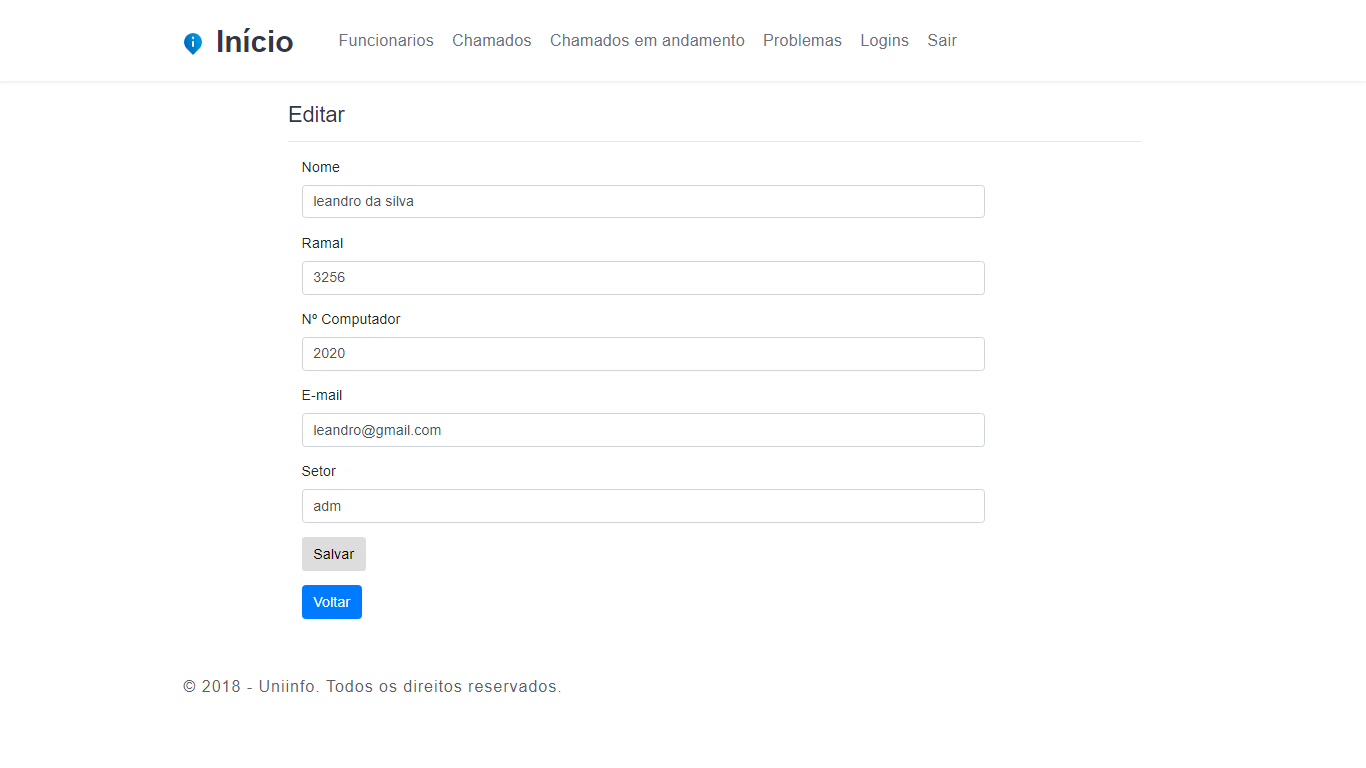
Fonte – Uniinfo 2018

Imagem 7 – Cadastrar Funcionário Pagina web administrador



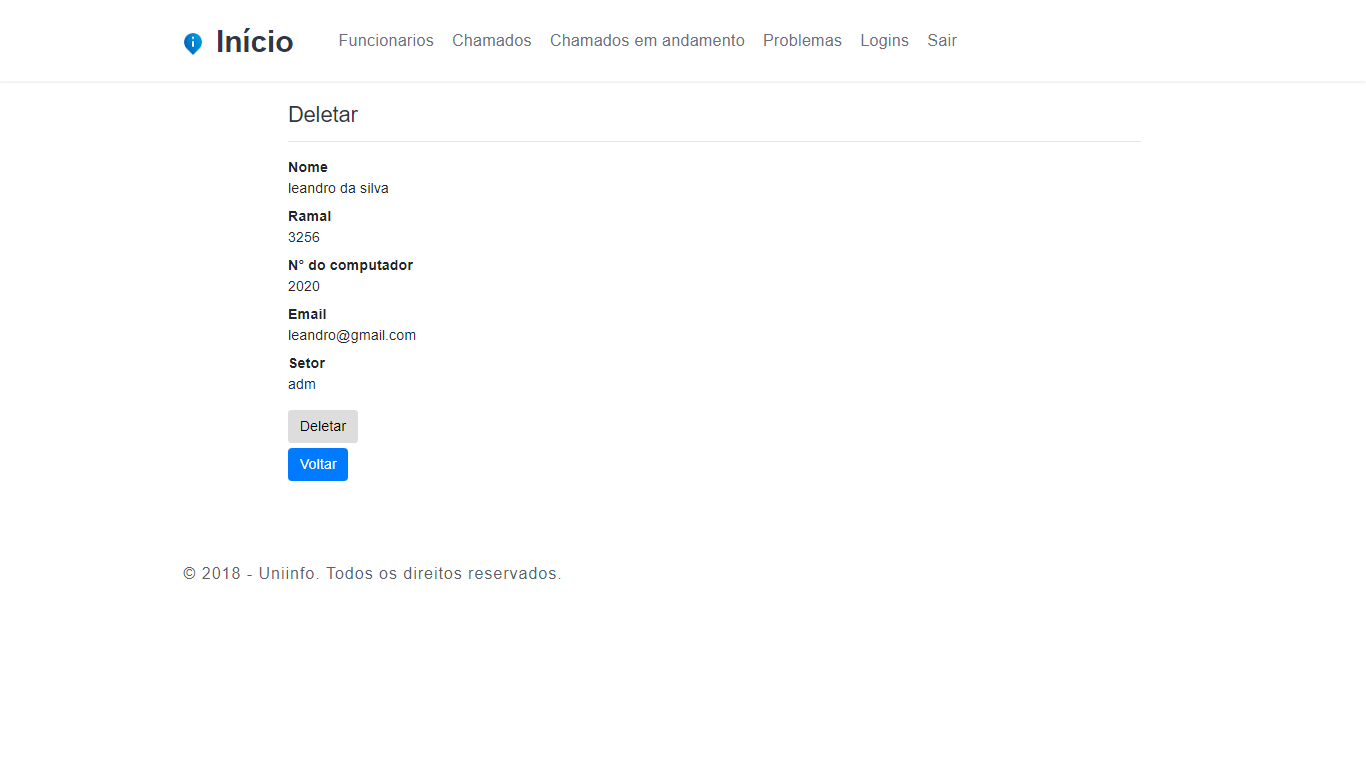
Fonte – Uniinfo 2018

Imagem 8 – Tela Editar funcionário Pagina web administrador



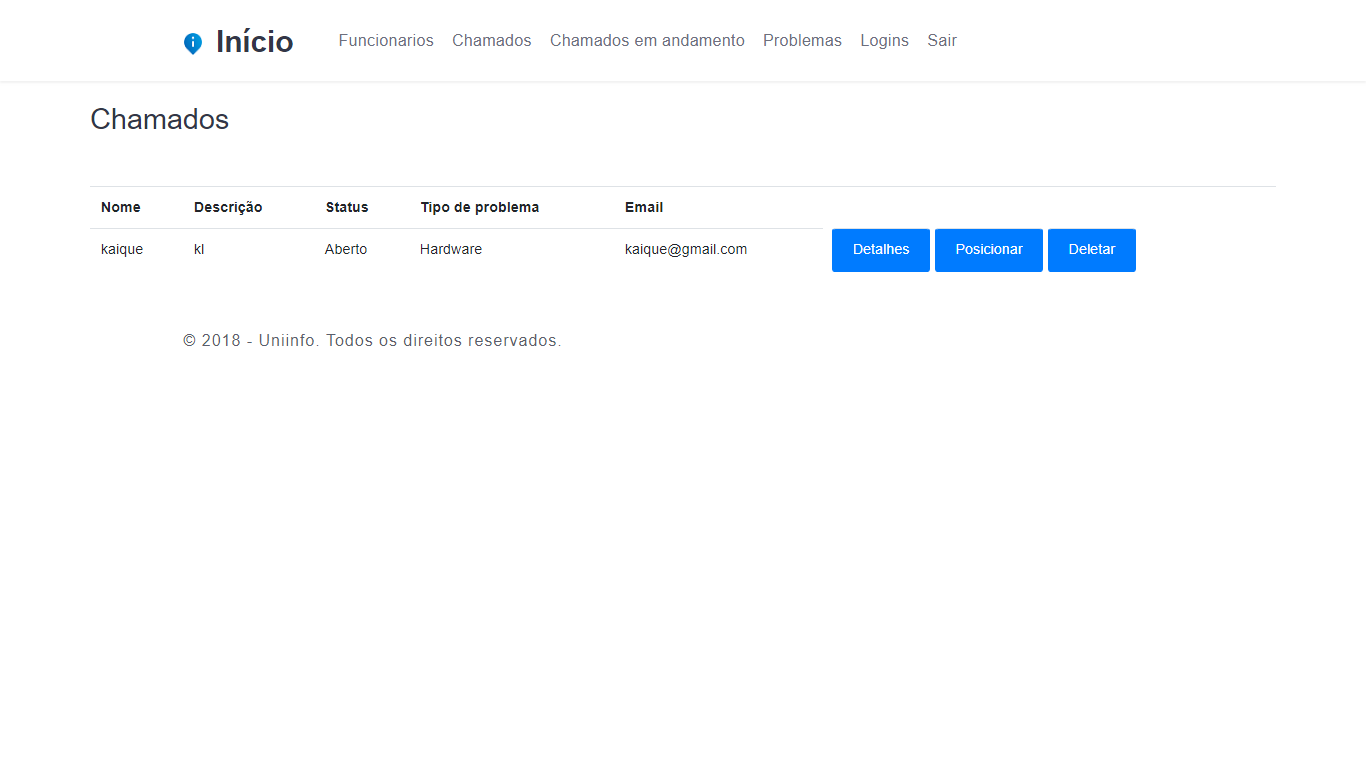
Fonte – Uniinfo 2018

Imagem 9 – Tela Deletar funcionário Pagina web administrador



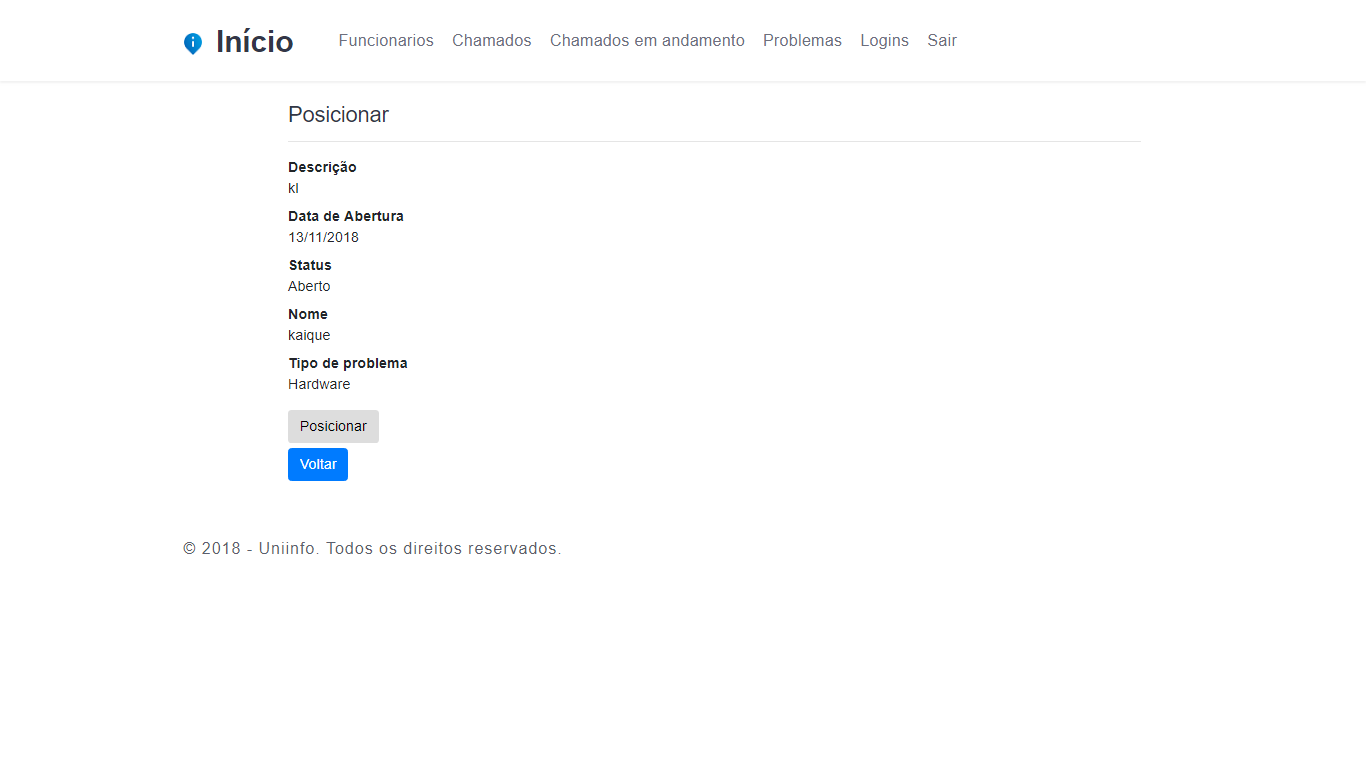
Fonte – Uniinfo 2018

Imagem 10 – Tela Chamados Pagina web administrador



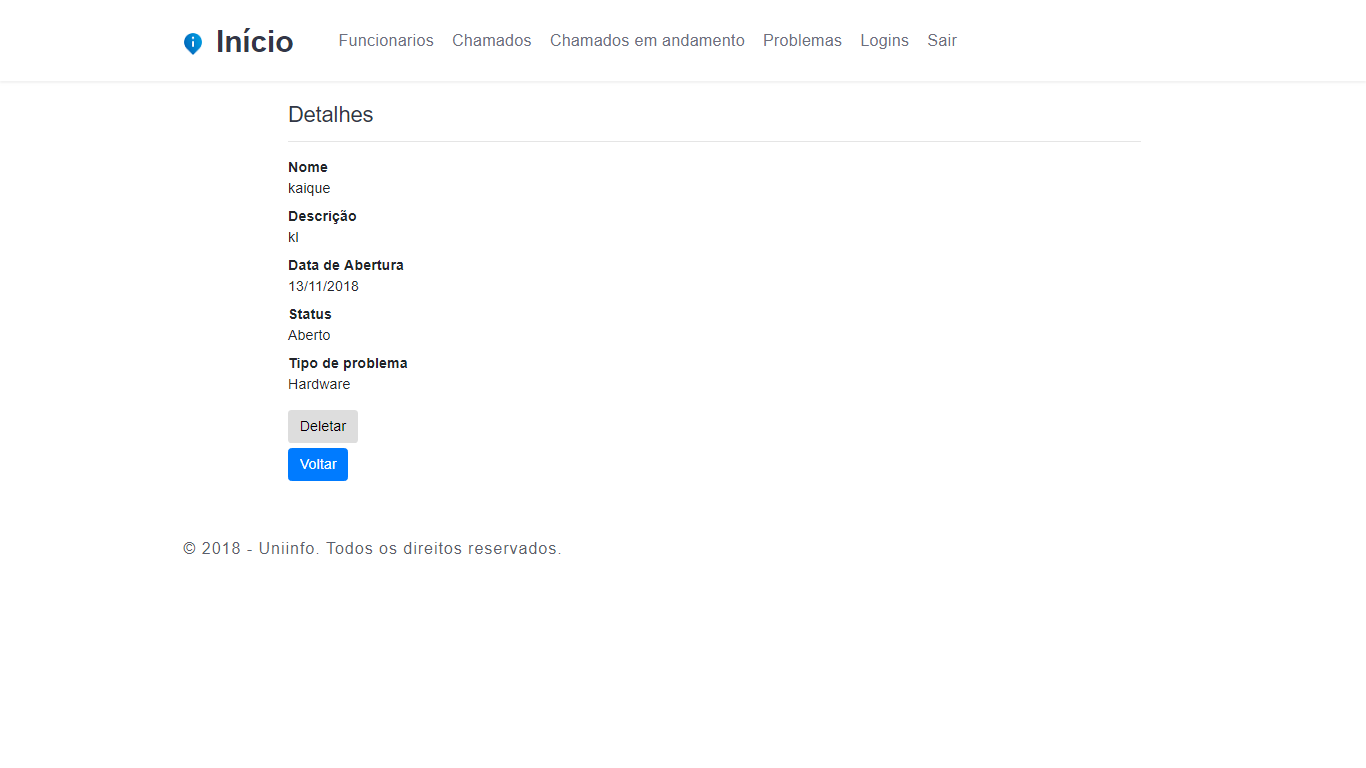
Fonte – Uniinfo 2018

Imagem 11 – Tela Posicionar Chamado Pagina web administrador



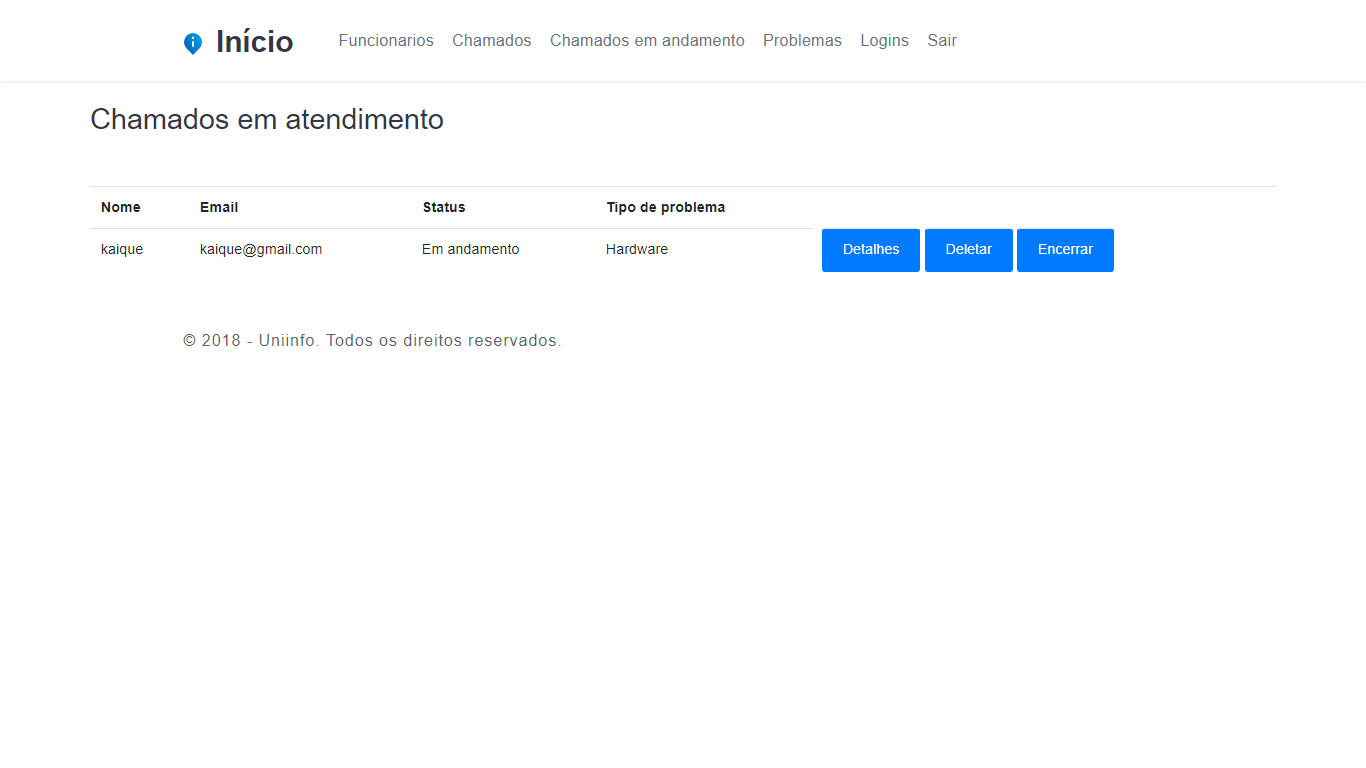
Fonte – Uniinfo 2018

Imagem 12 – Tela Detalhe do chamado Pagina web administrador



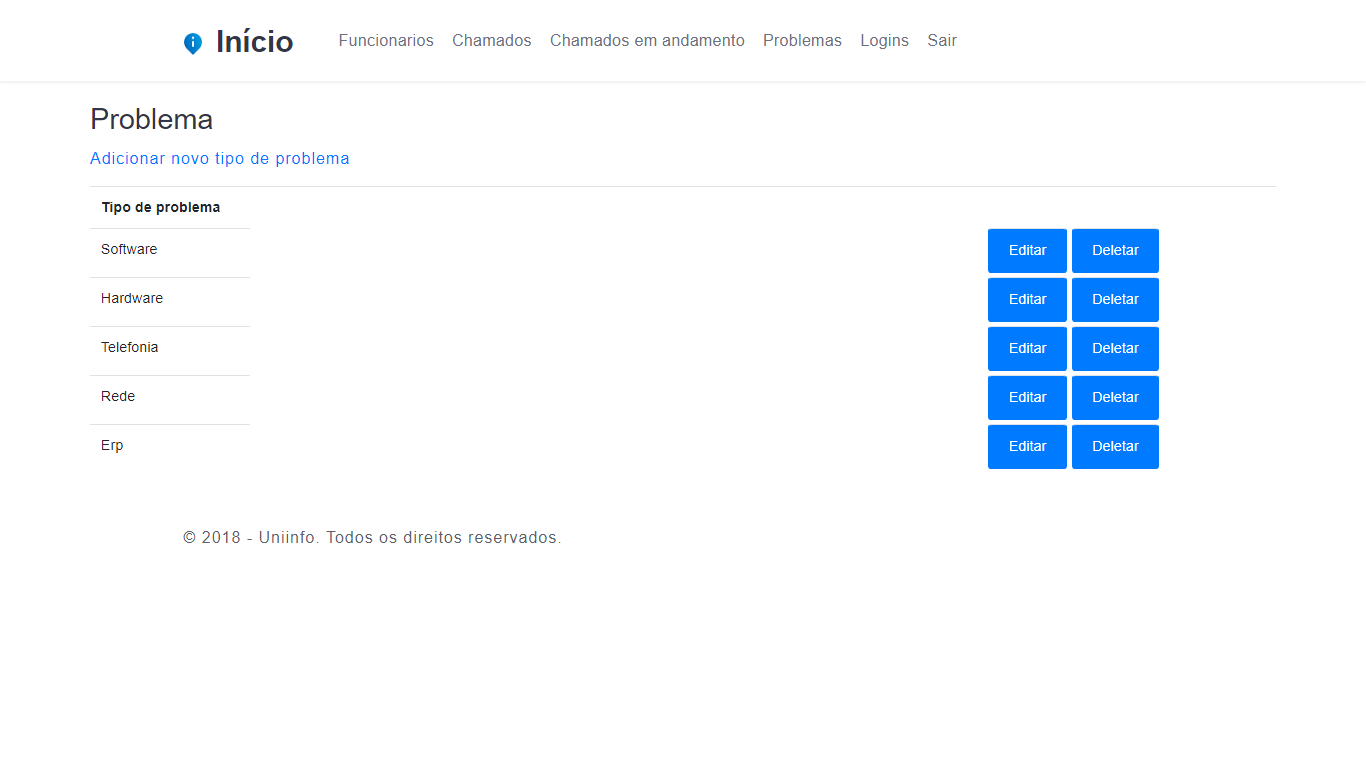
Fonte – Uniinfo 2018

Imagem 13 – Tela Chamado em atendimento Pagina web administrador



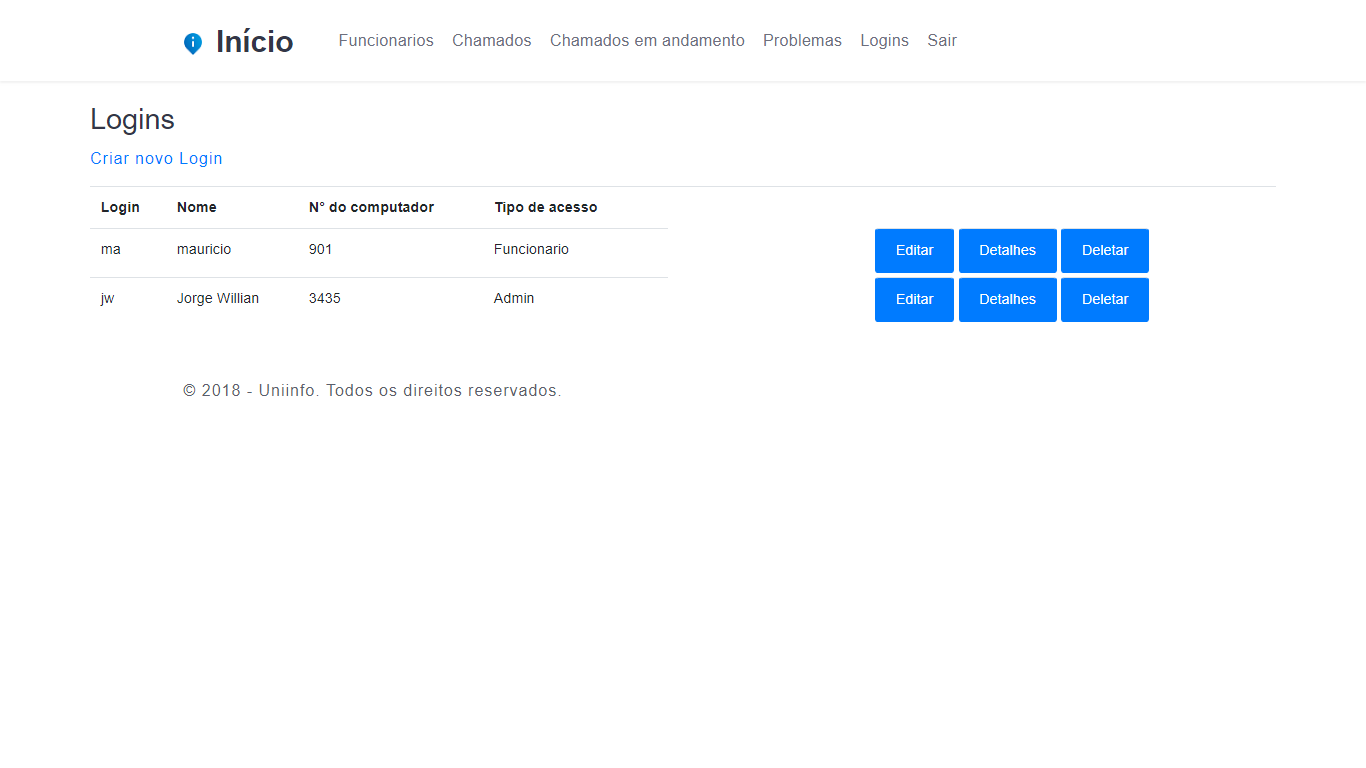
Fonte – Uniinfo 2018

Imagem 14 – Problemas Pagina web administrador



Fonte – Uniinfo 2018

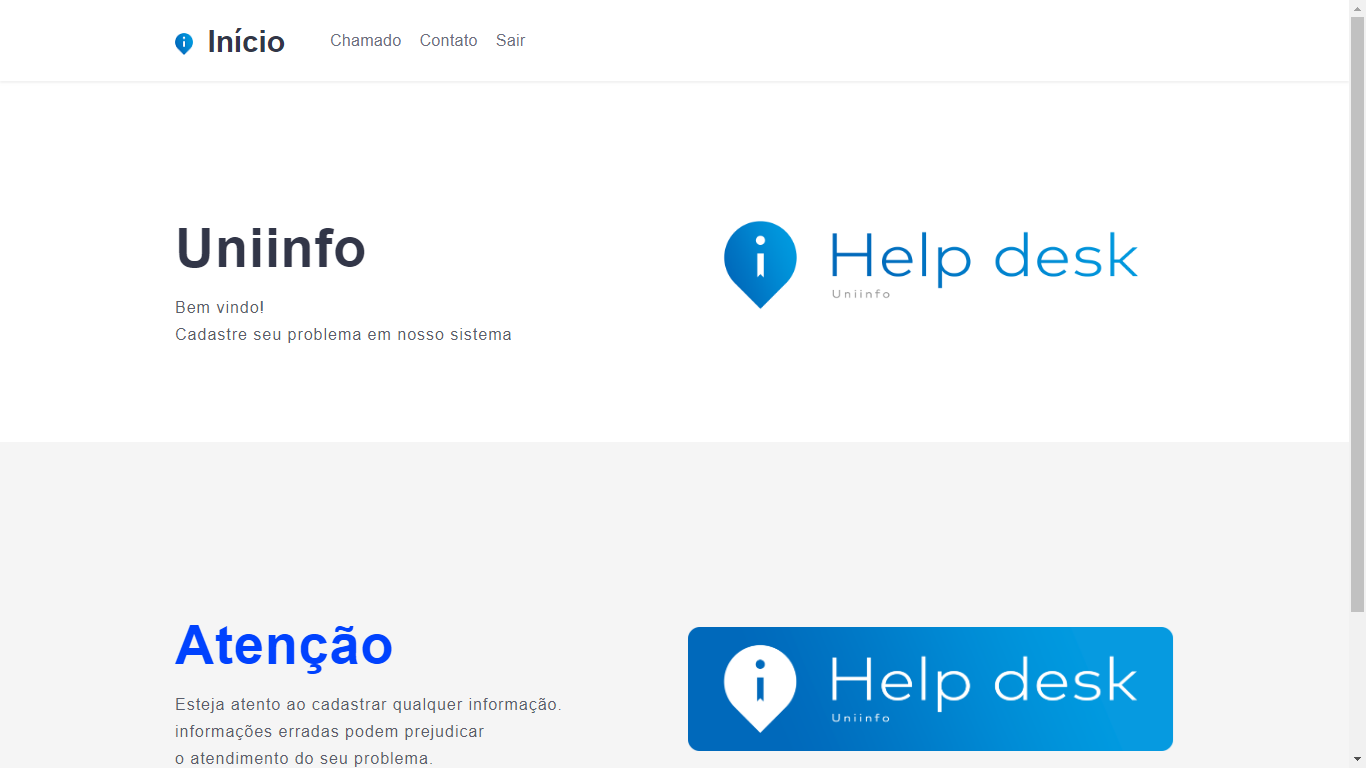
Imagem 15 – Tela Loguins Pagina web administrador



Fonte – Uniinfo 2018

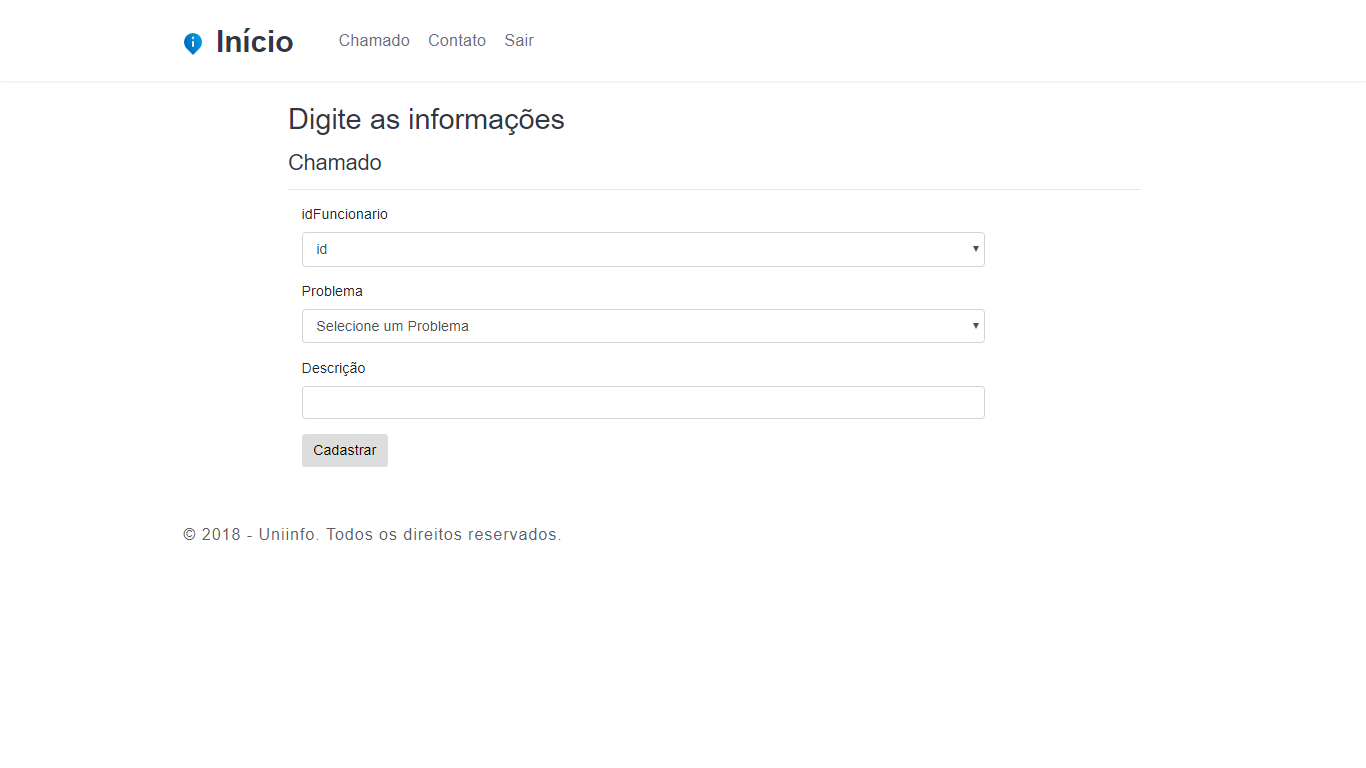
### Imagens das telas da plataforma web como Usuário

Imagem 16 – Tela inicio Pagina web usuario



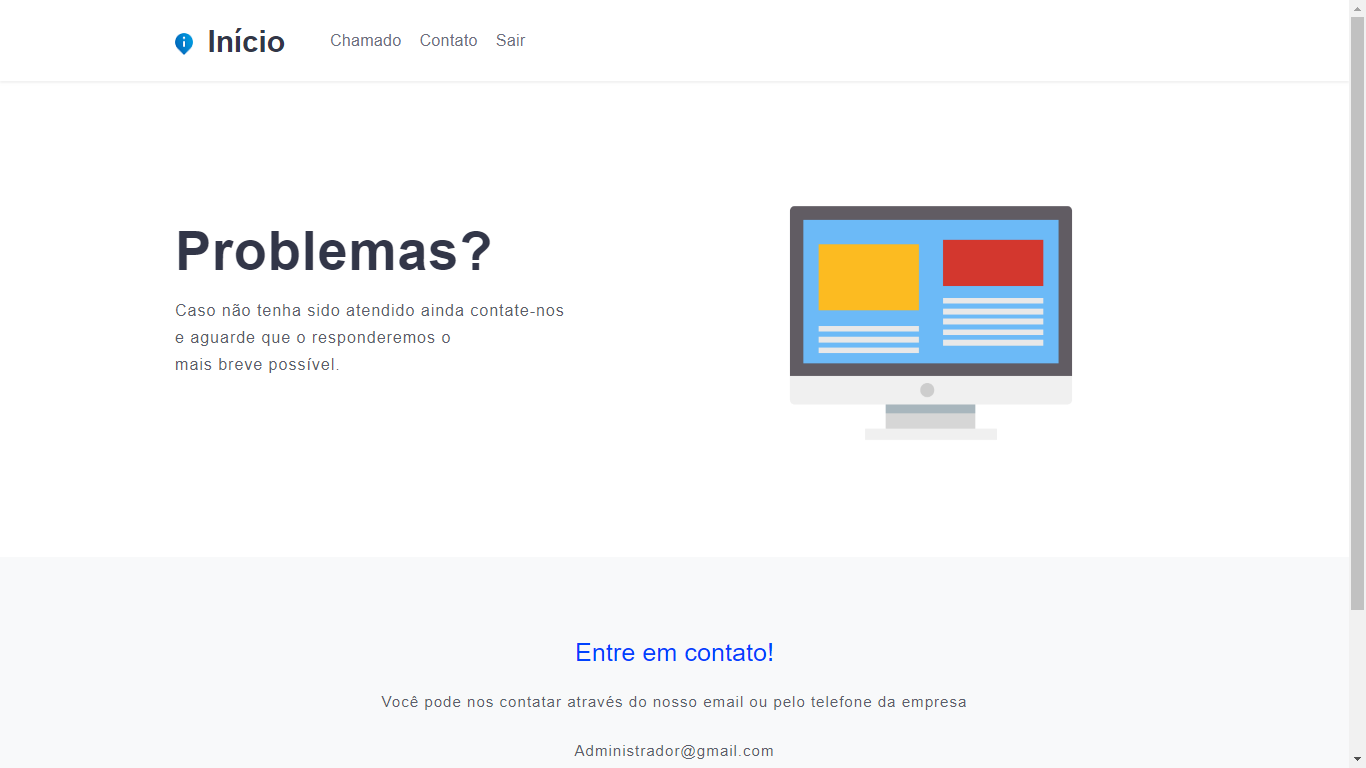
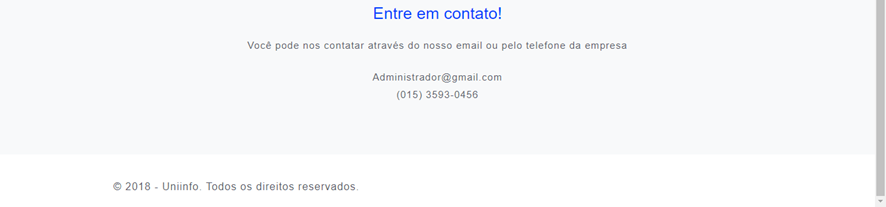
Fonte – Uniinfo 2018

Imagem 17 – Tela de abertura de chamado Pagina web usuário



Fonte – Uniinfo 2018

Imagem 18 – Tela de contato Pagina web usuário



Fonte – Uniinfo 2018

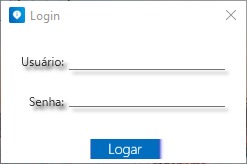
# HELP DESK DESKTOP

O Help Desk (Módulo Desktop da Uniinfo) foi desenvolvido com acesso apenas à logins cadastrados no sistema. Dessa forma, os administradores conseguem ter um controle absoluto de todos os chamados (Abertos, Em Andamento e Finalizados), juntamente com o sistema funcionando paralelamente com o suporte Web, onde todos os chamados são feitos.

Para que haja o funcionamento efetivo do Help Desk (Desktop) junto com o sistema Web, é necessário que o ASP.NET esteja online, já que o mesmo “off-line” exibirá uma mensagem de erro: “O servidor está com erro”, visto que o webservice está no ASP.NET.

## Imagens das telas desktop

Imagem 19 – Tela de Loguin Desktop



Fonte – Uniinfo 2018

Imagem 20 – Tela de consulta de Chamados Desktop



Fonte – Uniinfo 2018

# WEB SERVICE

Web Service é um sistema de software responsável por proporcionar a interação entre diversas máquinas através de uma rede. Para possibilitar uma interface descrita em um formato específico, WSDL (Web Services Description Language), permite que os sistemas interajam com o Web Service usando a interface e enviando mensagens SOAP (Simple Object Access Protocol) ou utilizando outros protocolos. As mensagens SOAP são basicamente documentos XML. O WSDL é uma linguagem para descrever o Web Services, é como um índice dos métodos disponíveis. O Web Service tem como o objetivo criar padrões comuns para conteúdo da Web, apoiada por grandes empresas como a Microsoft, IBM, HP entre outras.

# BANCO DE DADOS

Uma das definições de banco de dados afirma que se trata de uma coleção de informações que se relacionam de modo que criem algum sentido, isto é, é uma estrutura bem organizada de dados que permite a extração de informações

Modelagem de dados é adaptar um processo de um ambiente para o outro. Neste caso, dados. É através de sua junção ou de um determinado contexto, que o dado se transforma em informação. Então, na modelagem de dados, adaptamos os dados de um ambiente "físico" para o "computacional".

Sendo assim, existem três modelos necessários para a Modelagem do Banco de Dados: Conceitual, Lógico e Físico.

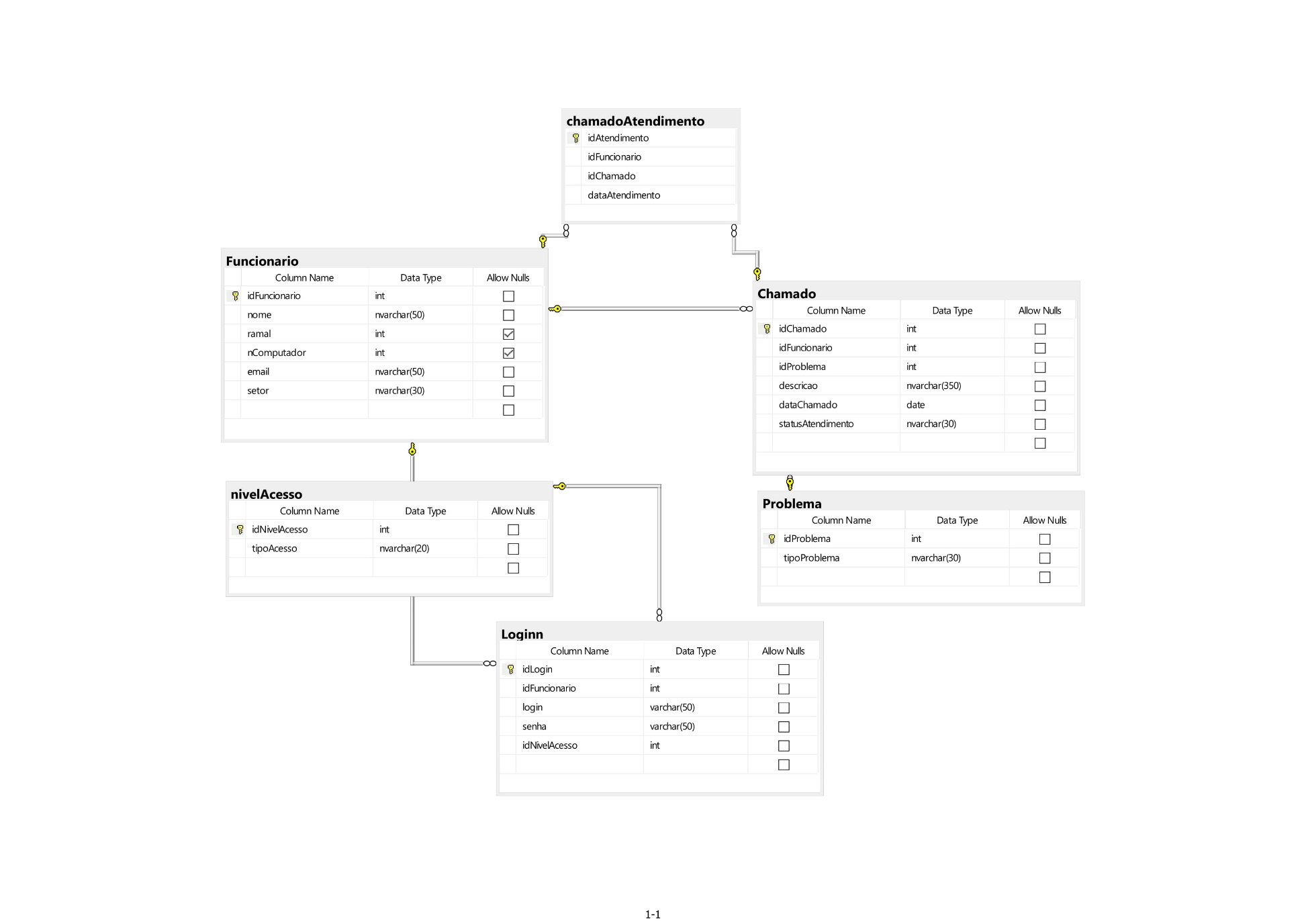
No modelo conceitual são levantados os requisitos dos usuários. O que ele deseja para o seu sistema.

No modelo Lógico o conceito lógico, ou seja, a representação gráfica dos requisitos, com apenas o que é realmente essencial, definindo as tabelas, os campos, os tipos de valores.

No modelo físico inclui a análise das características e recursos necessários para armazenamento e manipulação das estruturas de dados (estrutura de armazenamento, endereçamento, acesso e alocação física), a fim de criar as tabelas, estruturas e ligações projetadas até então e finalmente criar o banco de dados.

## Modelo lógico

Imagem 21 – Esquema Lógico Banco de Dados



Fonte – Uniinfo 2018

# DESENVOLVIMENTO DE SOFTWARE ÁGIL COM SCRUM

Scrum é um processo de desenvolvimento iterativo e incremental para construir e gerenciar projetos de software, onde cada parte do software é concluída em curto espaço de tempo. No SCRUM existem papéis bem definidos e diversas etapas que devem ser cumpridas em prazos estipulados visando entregar o produto de forma rápida e que ao mesmo tempo em que atenda as expectativas do cliente

No Scrum, os projetos são divididos em ciclos (tipicamente mensais) chamados de Sprints. O Sprint representa um Time Box dentro do qual um conjunto de atividades deve ser executado. Metodologias ágeis de desenvolvimento de software são iterativas, ou seja, o trabalho é dividido em iterações, que são chamadas de Sprints no caso do Scrum.

As funcionalidades a serem implementadas em um projeto são mantidas em uma lista que é conhecida como Product Backlog. No início de cada Sprint, faz-se um Sprint Planning Meeting, ou seja, uma reunião de planejamento na qual o Product Owner prioriza os itens do Product Backlog e a equipe seleciona as atividades que ela será capaz de implementar durante o Sprint que se inicia.

Ao final de um Sprint, a equipe apresenta as funcionalidades implementadas em uma Sprint Review Meeting. Finalmente, faz-se uma Sprint Retrospective, O Sprint Retrospective ocorre ao final de um Sprint e serve para identificar o que funcionou bem, o que pode ser melhorado e que ações serão tomadas para melhorar. e a equipe parte para o planejamento do próximo Sprint. Assim reinicia-se o ciclo.

## Product Backlog (Backlog do Produto)

O Product Backlog é o escopo do Produto. Neste artefato contém todo o escopo que deve ser produzido, recomenda-se ser a única fonte de requisitos do produto, para realmente poder ser a referência maior do time Scrum.

O Product Backlog é uma lista contendo todas as funcionalidades desejadas para um produto. O conteúdo desta lista é definido pelo Product Owner. O Product Backlog não precisa estar completo no início de um projeto. Pode-se começar com tudo aquilo que é mais óbvio em um primeiro momento. Com o tempo, o Product Backlog cresce e muda à medida que se aprende mais sobre o produto e seus usuários.

### Levantamento dos requisitos.

Nosso sistema Help-desk está sendo desenvolvido com base no uso profissional e em situações reais de uma empresa importadora de Correias e Mangueiras industriais localizada em Sorocaba O sistema deve atender as seguintes situações categorizadas: Hardware, Software, Sistema/ERP e Telefonia, cada uma delas com subcategorias que expandem suas informações e com detalhes para que o técnico possa efetuar o reparo com mais agilidade.

As funcionalidades que deverão conter no sistema para que o processo de ações seja ágil e organizado são: Prioridade, Status e situação.

Prioridade é o que diversifica o nível de precedência de um problema onde baixo tem um Sla (dia útil) de 3 dias, médio de 1 dia e alto de 4 horas úteis

Status indica em que fase de operação se encontra cada problema registrado nas chamadas, divididos em: aberto, em andamento e encerrado.

Situação valida se uma operação específica está sendo executada por algum técnico, situações não atribuídas não podem conter status em andamento ou encerrado.

“A engenharia de requisitos fornece um mecanismo adequado para entender o que o cliente deseja, analisar as necessidades, avaliar a exequibilidade, negociar uma solução razoável, especificar a solução de maneira não ambígua, validar a especificação e administrar os requisitos à medida que eles são transformados num sistema em operação”(PRESSMAN, 2002, p. 250)

## Sprint backlog

O Sprint Backlog é uma lista de tarefas que o Scrum Team se compromete a fazer em um Sprint. Os itens do Sprint Backlog são extraídos do Product Backlog, pela equipe, com base nas prioridades definidas pelo Product Owner e a percepção da equipe sobre o tempo que será necessário para completar as várias funcionalidades. Dessa forma com os requisitos já levantados, as prioridades foi o desenvolvimento de uma plataforma em asp.net junto com um web service e um sistema de consulta em WPF para desktop que atendesse as necessidades mínimas de funcionalidades do sistema.

# UML

UML é uma linguagem que define uma série de artefatos que nos ajuda na tarefa de modelar e documentar os sistemas orientados a objetos que desenvolvemos, é a parte anterior ao projeto, muitas vezes é até mesmo subestimada por ser algo “simples”, mas na verdade é a idealização do projeto em modelos relevantes a construção do software.

**Linguagem de modelagem unificada (UML)** é uma linguagem visual para documentação de projeto e padrões de software, podendo ser aplicada em várias áreas diferentes. É uma relação de documentação e expressões, comportamentos e ideias numa notação que é fácil de entender e eficiente para se escrever. UML é visual, quase tudo nela tem uma representação gráfica facilitando o entendimento de todos os envolvidos. Em geral, um modelo UML, é feito de um ou mais diagramas.

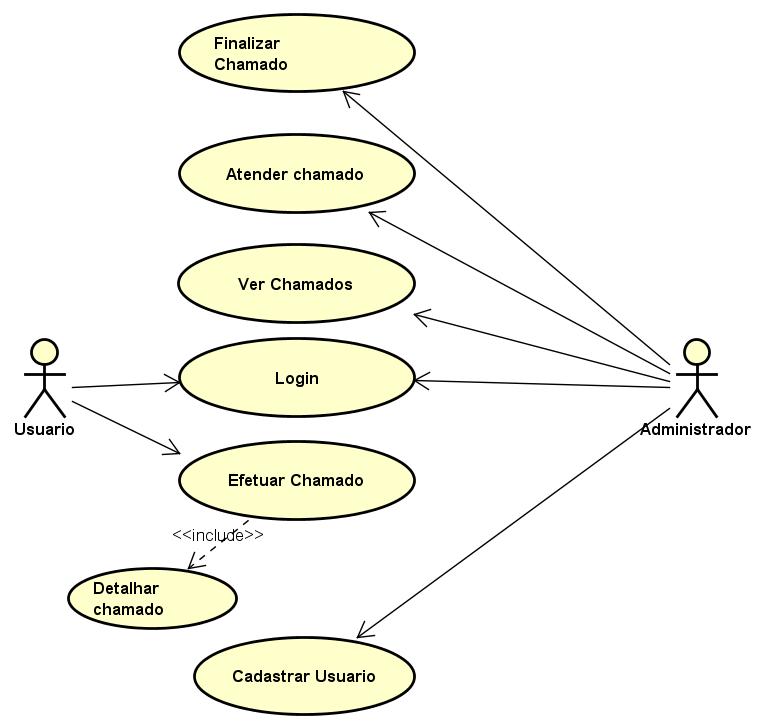
Um diagrama representa graficamente coisas e as relações entre essas coisas. Essas coisas podem ser representações de objetos do mundo real, construções de software ou uma descrição de comportamento de algum outro objeto. Aprender a aplicar a UML pressupõe o entendimento de três elementos, Itens, relacionamentos e diagramas.

## Diagrama de Caso de Uso

Esse diagrama documenta o que o sistema faz do ponto de vista do usuário. Em outras palavras, ele descreve as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários do mesmo sistema de uma forma que qualquer pessoa possa entender o que deverá acontecer no sistema. Nesse diagrama os detalhes são superficiais, ou seja, não define em detalhes técnicos como o sistema faz. Um caso de uso tem basicamente quatro partes, ator, cenário, caso de uso e comunicação.

O diagrama do caso de uso é a modelagem do caso de uso e um diagrama com desenhos e representações que exemplificam com praticidade quem são os atores e quais suas interações com o sistema uma vez definido o cenário

Imagem 22 – Diagrama de caso de Uso



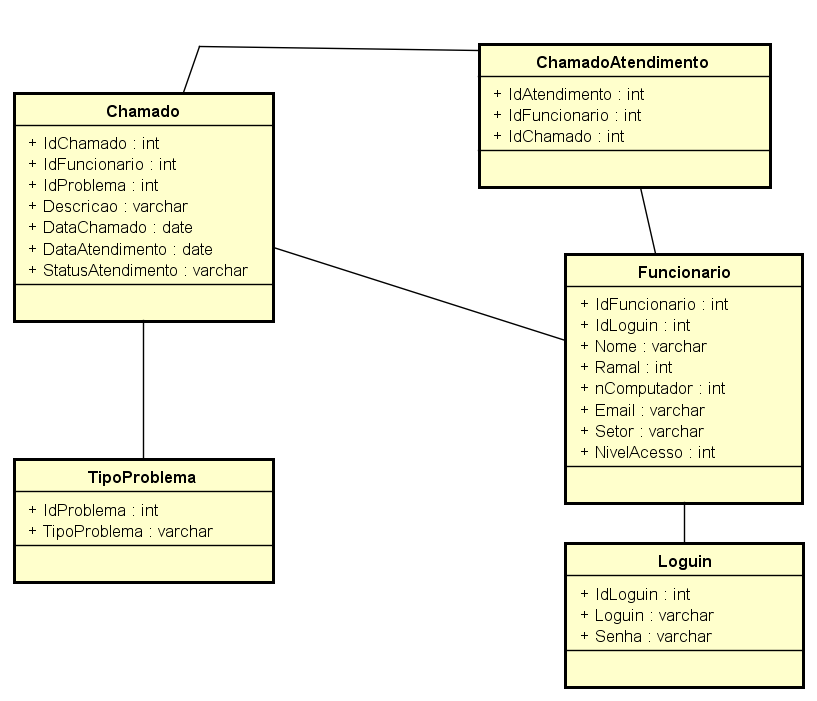
Fonte – Uniinfo 2018

## Diagrama de Classe

O diagrama de classes é utilizado na construção do modelo de classes desde o nível de análise até o nível de especificação. Cada classe tem por definição, Nome, Atributos e Métodos.

Uma classe é representada por uma “caixa” com, no máximo, três compartimentos exibidos. No primeiro compartimento (de cima para baixo) é exibido o nome da classe. Por convenção, esse nome é apresentado no singular e com as palavras componentes começando por maiúsculas. No segundo compartimento, são declarados os atributos, que correspondem às informações que um objeto armazena. Finalmente, no terceiro compartimento, são declaradas as operações, que correspondem às ações que um objeto sabe realizar.

Imagem 23 – Diagrama de Classe abrir e atender chamados

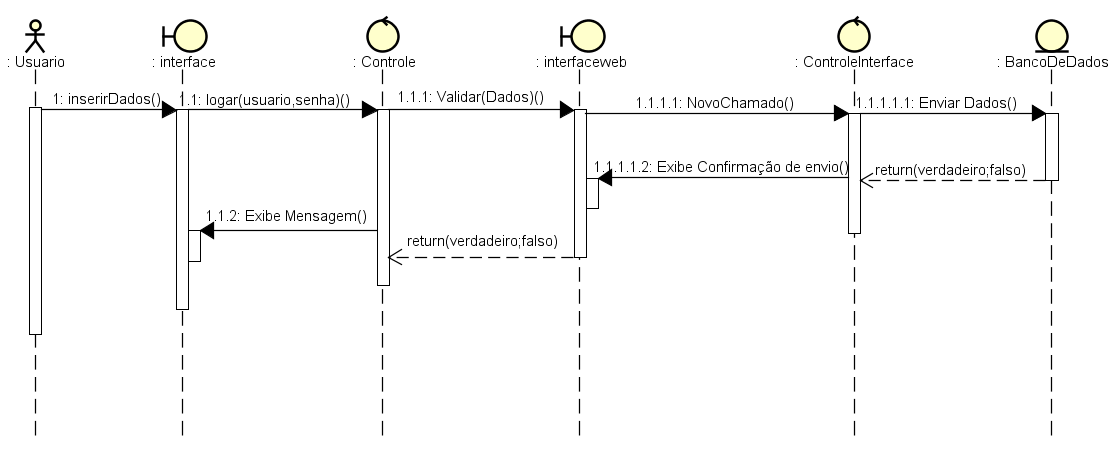


Fonte – Uniinfo 2018

## Diagrama de sequência

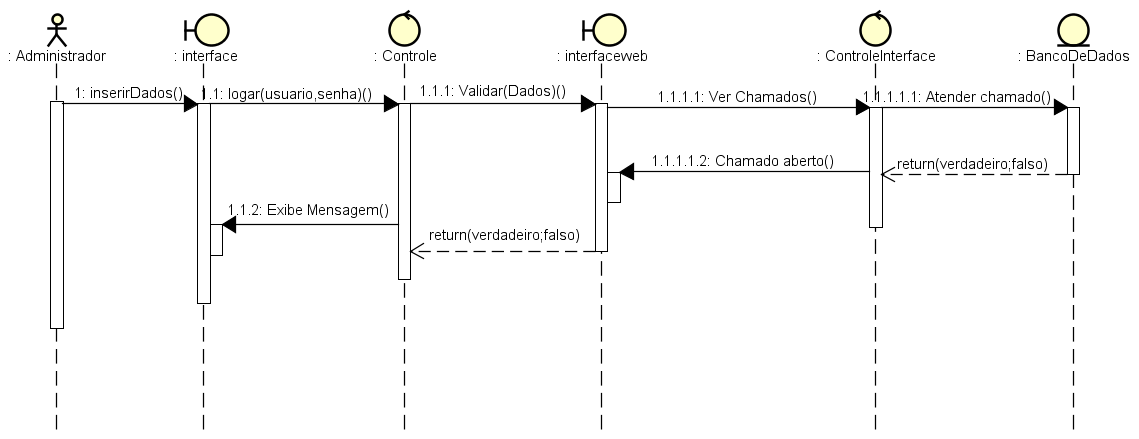
O diagrama de sequência visa fornecer visualmente a cadeia de processos de um caso de uso, mostrando passo a passo as interações dos objetos entre si em determinado cenário, desde o início até o fim do processo.

Imagem 24 – Diagrama de Sequência abertura de chamado



Fonte – Uniinfo 2018.

Imagem 25 – Diagrama de Sequência atendimento de chamado



Fonte – Uniinfo 2018.

# **PADRÕES DE QUALIDADE - ISO/IEC 9126 (NBR 13596)**

Uma norma que tem como objetivo avaliar a qualidade, bem como as suas características, subcaracterísticas e atributos. Este é o objetivo da ISO 9126, mais especificamente a NBR 13596, que padroniza a avaliação da qualidade do software. Um produto final com qualidade é uma necessidade, uma vez que desde o levantamento de requisitos até a fase de produção, os fatores que atendem ao que o cliente precisa devem estar em evidência, bem como o feedback para melhoria contínua dos processos da organização.

A norma ISO/IEC 9126, ou conjunto de normas que tratam deste assunto no âmbito da ISO, estabelece um modelo de qualidade com os componentes de Processo de desenvolvimento e Produto, cujo no Processo de desenvolvimento a qualidade afeta a qualidade do produto de software gerado e é influenciado pela natureza do produto desenvolvido, já o produto é

Compreendendo os atributos de qualidade do produto (sistema) de software. Estes atributos de qualidade podem ser divididos entre atributos internos e externos. Estes se diferenciam pela forma como são aferidos (interna ou externamente ao produto de software) e em conjunto compõem a qualidade do produto de software em si.

A norma 9126 foca na qualidade do produto de software, propondo Atributos de Qualidade, distribuídos em seis características principais, com cada uma delas divididas em subcaracterísticas: Funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade. Tendo em vista que estes são os requisitos exigidos pelos clientes e atendidos pelos fornecedores de Softwares. Seguindo estes, o resultado não poderá ser ruim. Teremos pontos positivos tanto para quem produz o software, quanto para quem utiliza o software, evitando assim tais gastos elevadíssimos gerados por má qualidade (Sem que algum ou todos os requisitos de Qualidade sejam atendidos).

Desta maneira o Help Desk Uniinfo foi desenvolvido buscando atender a todos critérios envolvidos na norma, para que assim, seja entregue um produto de alta qualidade para o usuário final.

# **CONCLUSÃO**

Conclui-se que o projeto Help Desk Uniifo é essencial para empresas onde existe a necessidade de soluções rápidas para problemas relacionados ao departamento de tecnologia da informação. O objetivo da equipe Uniinfo foi desenvolver um software ágil e eficaz, que por sua conta tem a função de reparar e auxiliar usuários com problemas, sejam eles, de softwares, de hardwares ou de telefonia, sendo assim, o projeto foi executado de maneira solida e consequentemente funcional. Por fim exaltamos todo estudo e conhecimento gerado por atividades do semestre e extracurriculares, de maneira que reconhecemos a extrema importância de um software Help Desk para empresas desde de pequeno a grande porte.

# 

# REFERÊNCIAS BIBLIOGRÁFICAS

(Aula 1) Prática - UML Diagrama de classes – (Atributos, Operações e Herança). Soares, Sthefane. Youtube. 02/102014 12min2s. Internet, disponivel em: <<https://www.youtube.com/watch?v=9Wdjuds5vPQ>>. Acessado 02/09/18.

Aula1 - Modelagem UML - Diagrama de Caso de Uso, Classe e Sequência. Brenner, Gabriel. Youtube. 21/06/2015 41min58s. Disponível em: <<https://www.youtube.com/watch?v=pCCqiJDLr4Y>> Acessado em 20/09/18.

.

Microsoft – Diagramas de sequência UML: referência. Microsoft, internet 2018. Disponivel em: <<https://msdn.microsoft.com/pt-br/library/dd409377.aspx>>. Acessado em 13/10/18.

Ribeiro, Leandro - O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML. DEVMEDIA, internet 2012. Disponivel em: <<https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>>. Acessado em 13/010/18.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR ISO-9126:* Engenharia de software. Rio de Janeiro, 2003.

Introdução ao Visual Studio. Disponível em:<<https://msdn.microsoft.com/pt-br/library/ms165079.aspx>>. Acessado em 11/11/18.

Introdução ao Banco de Dados SQL Server. Disponível em <<https://www.infoescola.com/informatica/sql-server/>>. Acessado em 11/11/18.

Introdução sobre a Linguagem de Programação CSharp. Disponível em: <<https://www.devmedia.com.br/introducao-a-linguagem-csharp/27711>>. Acessado em 11/11/18.

Resumo sobre modelagem de Banco de Dados. Disponível em <<http://aitibrasil.blogspot.com/2012/10/resumo-sobre-modelagem-de-dados.html.>>. Acessado em 11/11/18.

Introdução ao Diagrama de Classes. Disponível em <<https://www.devmedia.com.br/diagrama-de-classes-uml/12251>>. Acesso 11/11/18.

Introdução ao Dot.Net. Disponível em <<https://docs.microsoft.com/pt-br/dotnet/core/get-started?tabs=windows>.> Acessado em 12/11/18.

Guia de Desenvolvimento do .NET Framework. Disponível em: < <https://docs.microsoft.com/pt-br/dotnet/framework/development-guide>> Acessado em 11/11/18.

Resumo sobre Web Service e Funcionalidades. Disponível em <<https://pt.stackoverflow.com/questions/90315/o-que-é-um-web-service>>. Acessado em 12/11/18.

FOWLER, M. UML Essencial. 3ª Ed., São Paulo: Bookman, 2005

CRUZ, F. Scrum e PMBOK unidos no Gerenciamento de Projetos. Rio de Janeiro: Brasport, 2013.

PRESSMAN, R. Engenharia de Software: Uma Abordagem Profissional. 7ª Ed.,

São Paulo: McGraw-Hill, 2011.

# APÊNDICE

# APÊNDICE A - CÓDIGO FONTE ASP.NET

**APPSTART**

using System.Web;

using System.Web.Optimization;

namespace UniinfoAsp

{

public class BundleConfig

{

// Para obter mais informações sobre o agrupamento, visite https://go.microsoft.com/fwlink/?LinkId=301862

public static void RegisterBundles(BundleCollection bundles)

{

bundles.Add(new ScriptBundle("~/bundles/jquery").Include(

"~/Scripts/jquery-{version}.js"));

bundles.Add(new ScriptBundle("~/bundles/jqueryval").Include(

"~/Scripts/jquery.validate\*"));

// Use a versão em desenvolvimento do Modernizr para desenvolver e aprender. Em seguida, quando estiver

// pronto para a produção, utilize a ferramenta de build em https://modernizr.com para escolher somente os testes que precisa.

bundles.Add(new ScriptBundle("~/bundles/modernizr").Include(

"~/Scripts/modernizr-\*"));

bundles.Add(new ScriptBundle("~/bundles/bootstrap").Include(

"~/Scripts/bootstrap.js",

"~/Scripts/respond.js"));

bundles.Add(new StyleBundle("~/Content/css").Include(

"~/Content/bootstrap2.css",

"~/Content/style2.css"));

}

}

}

using System.Web;

using System.Web.Mvc;

namespace UniinfoAsp

{

public class FilterConfig

{

public static void RegisterGlobalFilters(GlobalFilterCollection filters)

{

filters.Add(new HandleErrorAttribute());

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Web.Routing;

namespace UniinfoAsp

{

public class RouteConfig

{

public static void RegisterRoutes(RouteCollection routes)

{

routes.IgnoreRoute("{resource}.axd/{\*pathInfo}");

routes.MapRoute(

name: "Default",

url: "{controller}/{action}/{id}",

defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }

);

}

}

}

**CONTROLLERS**

using System.Linq;

using System.Web.Mvc;

using System.Web.Security;

using UniinfoAsp.Models;

namespace UniinfoAsp.Controllers

{

public class AccountController : Controller

{

UnipEntities db = new UnipEntities();

public ActionResult Login()

{

ViewBag.Message = "";

return View();

}

[HttpPost]

public ActionResult Login(Loginn log)

{

var result = db.Loginns.Where(a => a.login == log.login && a.senha == log.senha).ToList();

if (result.Count() > 0)

{

Session["idLogin"] = result[0].idLogin;

FormsAuthentication.SetAuthCookie(result[0].login, false);

//If admin

if (result[0].idNivelAcesso == 1)

{

return RedirectToAction("../Admin/index");

}

//If Funcionario

if (result[0].idNivelAcesso == 2)

{

return RedirectToAction("../Func/index");

}

}

else

{

ViewBag.Message = "Erro, Usuario ou senha incorreto";

}

return View(log);

}

public ActionResult LogOut()

{

//Session["idLogin"] = 0;

//Session.Abandon();

FormsAuthentication.SignOut();

return RedirectToAction("Login");

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace UniinfoAsp.Controllers

{

public class AdminController : Controller

{

[Authorize(Roles = "Admin")]

public ActionResult Index()

{

return View();

}

}

}

using System;

using System.Data;

using System.Data.Entity;

using System.Linq;

using System.Net;

using System.Web.Mvc;

using UniinfoAsp.Models;

namespace UniinfoAsp.Controllers

{

[Authorize(Roles = "Admin")]

public class chamadoAtendimentoController : Controller

{

private UnipEntities db = new UnipEntities();

// GET: chamadoAtendimento

public ActionResult Index()

{

var chamadoAtendimentoes = db.chamadoAtendimentoes.Include(c => c.Chamado).Include(c => c.Funcionario);

var chamAtend = chamadoAtendimentoes.Where(ca => ca.Chamado.statusAtendimento.Equals("Em andamento"));

return View(chamAtend.ToList());

}

// GET: chamadoAtendimento/Details/5

public ActionResult Details(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

chamadoAtendimento chamadoAtendimento = db.chamadoAtendimentoes.Find(id);

if (chamadoAtendimento == null)

{

return HttpNotFound();

}

return View(chamadoAtendimento);

}

// GET: chamadoAtendimento/Delete/5

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

chamadoAtendimento chamadoAtendimento = db.chamadoAtendimentoes.Find(id);

if (chamadoAtendimento == null)

{

return HttpNotFound();

}

return View(chamadoAtendimento);

}

// POST: chamadoAtendimento/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

chamadoAtendimento chamadoAtendimento = db.chamadoAtendimentoes.Find(id);

db.chamadoAtendimentoes.Remove(chamadoAtendimento);

db.SaveChanges();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

public ActionResult Encerrar(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

chamadoAtendimento chamadoAtendimento = db.chamadoAtendimentoes.Find(id);

Chamado chamadoa = db.Chamadoes.Find(id);

if (chamadoAtendimento == null)

{

return HttpNotFound();

}

return View(chamadoAtendimento);

}

[HttpPost, ActionName("Encerrar")]

[ValidateAntiForgeryToken]

public ActionResult EncerrarStatus(int id)

{

var chamadoAtendimento = db.chamadoAtendimentoes.Find(id);

chamadoAtendimento.Chamado.statusAtendimento = "Encerrado";

db.Entry(chamadoAtendimento).Property(ca => ca.dataAtendimento).CurrentValue = DateTime.Now;

db.Entry(chamadoAtendimento.Chamado).Property(c => c.statusAtendimento).IsModified = true;

db.SaveChanges();

return RedirectToAction("Index");

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Entity;

using System.Linq;

using System.Net;

using System.Web;

using System.Web.Mvc;

using UniinfoAsp.Models;

namespace UniinfoAsp.Controllers

{

[Authorize(Roles = "Admin")]

public class ChamadoController : Controller

{

private UnipEntities db = new UnipEntities();

// GET: Chamado

public ActionResult Index()

{

var chamadoes = db.Chamadoes.Include(c => c.Funcionario).Include(c => c.Problema);

var cham = chamadoes.Where(c => c.statusAtendimento.Equals("Aberto"));

return View(cham.ToList());

}

// GET: Chamado/Details/5

public ActionResult Details(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Chamado chamado = db.Chamadoes.Find(id);

if (chamado == null)

{

return HttpNotFound();

}

return View(chamado);

}

[HttpPost, ActionName("Details")]

[ValidateAntiForgeryToken]

public ActionResult DetailsDeleteConfirmed(int id)

{

Chamado chamado = db.Chamadoes.Find(id);

db.Chamadoes.Remove(chamado);

db.SaveChanges();

return RedirectToAction("Index");

}

// GET: Chamado/Delete/5

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Chamado chamado = db.Chamadoes.Find(id);

if (chamado == null)

{

return HttpNotFound();

}

return View(chamado);

}

// POST: Chamado/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

Chamado chamado = db.Chamadoes.Find(id);

db.Chamadoes.Remove(chamado);

db.SaveChanges();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

public ActionResult Posicionar(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Chamado chamado = db.Chamadoes.Find(id);

if (chamado == null)

{

return HttpNotFound();

}

return View(chamado);

}

[HttpPost, ActionName("Posicionar")]

[ValidateAntiForgeryToken]

public ActionResult posicionarStatus(int id)

{

var chamado = db.Chamadoes.Find(id);

chamado.statusAtendimento = "Em andamento";

db.Chamadoes.Attach(chamado);

db.Entry(chamado).Property(c => c.statusAtendimento).IsModified = true;

db.SaveChanges();

chamadoAtendimento ca = new chamadoAtendimento();

ca.idChamado = chamado.idChamado;

ca.idFuncionario = chamado.Funcionario.idFuncionario;

db.chamadoAtendimentoes.Add(ca);

db.SaveChanges();

db.SaveChanges();

return RedirectToAction("Index");

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace UniinfoAsp.Controllers

{

[Authorize(Roles = "Funcionario")]

public class ContatoController : Controller

{

// GET: Contato

public ActionResult Index()

{

return View();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using UniinfoAsp.Models;

namespace UniinfoAsp.Controllers

{

[Authorize(Roles = "Funcionario")]

public class FuncController : Controller

{

UnipEntities Db = new UnipEntities();

public ActionResult Index()

{

return View();

}

public ActionResult CadastrarChamado()

{

ViewBag.funcionarioId = Db.Funcionarios.ToList();

//ViewBag.idFuncionario = new SelectList(Db.Funcionarios.Where(x => x.nome == User.Identity.Name), "idFuncionario", "nome");

ViewBag.problema = Db.Problemas.ToList();

return View();

}

[HttpPost]

public ActionResult CadastrarChamado(Chamado Cham)

{

Cham.dataChamado = System.DateTime.Now;

Cham.statusAtendimento = "Aberto";

Db.Chamadoes.Add(Cham);

Db.SaveChanges();

return RedirectToAction("CadastrarChamado");

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Entity;

using System.Linq;

using System.Net;

using System.Web;

using System.Web.Mvc;

using UniinfoAsp.Models;

namespace UniinfoAsp.Controllers

{

[Authorize(Roles = "Admin")]

public class FuncionarioController : Controller

{

private UnipEntities db = new UnipEntities();

// GET: Funcionario

public ActionResult Index()

{

return View(db.Funcionarios.ToList());

}

// GET: Funcionario/Create

public ActionResult Create()

{

return View();

}

// POST: Funcionario/Create

// Para se proteger de mais ataques, ative as propriedades específicas a que você quer se conectar. Para

// obter mais detalhes, consulte https://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create([Bind(Include = "idFuncionario,nome,ramal,nComputador,email,setor")] Funcionario funcionario)

{

if (ModelState.IsValid)

{

db.Funcionarios.Add(funcionario);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(funcionario);

}

//public JsonResult computadorCadastrado(int nComputador)

//{

// return Json(!db.Funcionarios.Any(f => f.nComputador == nComputador), JsonRequestBehavior.AllowGet);

//}

// GET: Funcionario/Edit/5

public ActionResult Edit(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Funcionario funcionario = db.Funcionarios.Find(id);

if (funcionario == null)

{

return HttpNotFound();

}

return View(funcionario);

}

// POST: Funcionario/Edit/5

// Para se proteger de mais ataques, ative as propriedades específicas a que você quer se conectar. Para

// obter mais detalhes, consulte https://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit([Bind(Include = "idFuncionario,nome,ramal,nComputador,email,setor")] Funcionario funcionario)

{

if (ModelState.IsValid)

{

db.Entry(funcionario).State = EntityState.Modified;

db.SaveChanges();

return RedirectToAction("Index");

}

return View(funcionario);

}

// GET: Funcionario/Delete/5

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Funcionario funcionario = db.Funcionarios.Find(id);

if (funcionario == null)

{

return HttpNotFound();

}

return View(funcionario);

}

// POST: Funcionario/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

Funcionario funcionario = db.Funcionarios.Find(id);

if (db.Chamadoes.Any(f => f.idFuncionario == funcionario.idFuncionario))

{

ViewBag.Message = "Se o usuário estiver relacionado a um chamado (Aberto, Em andamento, Encerrado) ele não poderá ser apagado";

}

else

{

db.Funcionarios.Remove(funcionario);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(funcionario);

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

}

//[Authorize(Roles = "")]

//public class error : Controller

//{

// public ActionResult erro()

// {

// return RedirectToAction("Account/Login");

// }

//}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace UniinfoAsp.Controllers

{

public class HomeController : Controller

{

public ActionResult Index()

{

return View();

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Entity;

using System.Linq;

using System.Net;

using System.Web;

using System.Web.Mvc;

using UniinfoAsp.Models;

namespace UniinfoAsp.Controllers

{

[Authorize(Roles = "Admin")]

public class LoginnController : Controller

{

private UnipEntities db = new UnipEntities();

// GET: Loginn

public ActionResult Index()

{

var loginns = db.Loginns.Include(l => l.Funcionario).Include(l => l.nivelAcesso);

return View(loginns.ToList());

}

// GET: Loginn/Details/5

public ActionResult Details(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Loginn loginn = db.Loginns.Find(id);

if (loginn == null)

{

return HttpNotFound();

}

return View(loginn);

}

[HttpPost, ActionName("Details")]

[ValidateAntiForgeryToken]

public ActionResult detailsDeleteConfirmed(int id)

{

Loginn loginn = db.Loginns.Find(id);

if (loginn.login.Equals("jw"))

{

ViewBag.Message = "Este login não pode ser apagado";

}

else

{

db.Loginns.Remove(loginn);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(loginn);

}

// GET: Loginn/Create

public ActionResult Create()

{

ViewBag.idFuncionario = new SelectList(db.Funcionarios, "idFuncionario", "nome");

ViewBag.idNivelAcesso = new SelectList(db.nivelAcessoes, "idNivelAcesso", "tipoAcesso");

return View();

}

// POST: Loginn/Create

// Para se proteger de mais ataques, ative as propriedades específicas a que você quer se conectar. Para

// obter mais detalhes, consulte https://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create([Bind(Include = "idLogin,idFuncionario,login,senha,idNivelAcesso")] Loginn loginn)

{

if (ModelState.IsValid)

{

db.Loginns.Add(loginn);

db.SaveChanges();

return RedirectToAction("Index");

}

ViewBag.idFuncionario = new SelectList(db.Funcionarios, "idFuncionario", "nome", loginn.idFuncionario);

ViewBag.idNivelAcesso = new SelectList(db.nivelAcessoes, "idNivelAcesso", "tipoAcesso", loginn.idNivelAcesso);

return View(loginn);

}

public JsonResult loginexistente(string login)

{

return Json(!db.Loginns.Any(l => l.login == login), JsonRequestBehavior.AllowGet);

}

// GET: Loginn/Edit/5

public ActionResult Edit(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Loginn loginn = db.Loginns.Find(id);

if (loginn == null)

{

return HttpNotFound();

}

ViewBag.idFuncionario = new SelectList(db.Funcionarios, "idFuncionario", "nome", loginn.idFuncionario);

ViewBag.idNivelAcesso = new SelectList(db.nivelAcessoes, "idNivelAcesso", "tipoAcesso", loginn.idNivelAcesso);

return View(loginn);

}

// POST: Loginn/Edit/5

// Para se proteger de mais ataques, ative as propriedades específicas a que você quer se conectar. Para

// obter mais detalhes, consulte https://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit([Bind(Include = "idLogin,idFuncionario,login,senha,idNivelAcesso")] Loginn loginn)

{

if (ModelState.IsValid)

{

db.Entry(loginn).State = EntityState.Modified;

db.SaveChanges();

return RedirectToAction("Index");

}

ViewBag.idFuncionario = new SelectList(db.Funcionarios, "idFuncionario", "nome", loginn.idFuncionario);

ViewBag.idNivelAcesso = new SelectList(db.nivelAcessoes, "idNivelAcesso", "tipoAcesso", loginn.idNivelAcesso);

return View(loginn);

}

// GET: Loginn/Delete/5

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Loginn loginn = db.Loginns.Find(id);

if (loginn == null)

{

return HttpNotFound();

}

return View(loginn);

}

// POST: Loginn/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

Loginn loginn = db.Loginns.Find(id);

if(loginn.login.Equals("jw"))

{

ViewBag.Message = "Este login não pode ser apagado";

}

else

{

db.Loginns.Remove(loginn);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(loginn);

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

}

}

using System.Data.Entity;

using System.Linq;

using System.Net;

using System.Web.Mvc;

using UniinfoAsp.Models;

namespace UniinfoAsp.Controllers

{

[Authorize(Roles = "Admin")]

public class ProblemaController : Controller

{

private UnipEntities db = new UnipEntities();

// GET: Problema

public ActionResult Index()

{

return View(db.Problemas.ToList());

}

// GET: Problema/Create

public ActionResult Create()

{

return View();

}

// POST: Problema/Create

// Para se proteger de mais ataques, ative as propriedades específicas a que você quer se conectar. Para

// obter mais detalhes, consulte https://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create([Bind(Include = "idProblema,tipoProblema")] Problema problema)

{

if (ModelState.IsValid)

{

db.Problemas.Add(problema);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(problema);

}

public JsonResult problemaExistente(string tipoProblema)

{

return Json(!db.Problemas.Any(p => p.tipoProblema == tipoProblema), JsonRequestBehavior.AllowGet);

}

// GET: Problema/Edit/5

public ActionResult Edit(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Problema problema = db.Problemas.Find(id);

if (problema == null)

{

return HttpNotFound();

}

return View(problema);

}

// POST: Problema/Edit/5

// Para se proteger de mais ataques, ative as propriedades específicas a que você quer se conectar. Para

// obter mais detalhes, consulte https://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit([Bind(Include = "idProblema,tipoProblema")] Problema problema)

{

if (ModelState.IsValid)

{

db.Entry(problema).State = EntityState.Modified;

db.SaveChanges();

return RedirectToAction("Index");

}

return View(problema);

}

// GET: Problema/Delete/5

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Problema problema = db.Problemas.Find(id);

if (problema == null)

{

return HttpNotFound();

}

return View(problema);

}

// POST: Problema/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

Problema problema = db.Problemas.Find(id);

if (db.Chamadoes.Any(c => c.idProblema == problema.idProblema))

{

ViewBag.Message = "Se o Problema estiver relacionado a um chamado ele não poderá ser apagado";

}

else

{

db.Problemas.Remove(problema);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(problema);

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace UniinfoAsp.Controllers

{

public class SobreController : Controller

{

// GET: Sobre

public ActionResult Index()

{

return View();

}

}

}

**DAL**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using UniinfoAsp.Models;

namespace UniinfoAsp.DAL

{

public class chamadoDAO

{

UnipEntities db = new UnipEntities();

public List<Con\_Chamado> Consultar()

{

var novo = db.Con\_Chamado.ToList();

return novo;

}

public List<Chamado> Consultar2()

{

var novo2 = db.Chamadoes.ToList();

return novo2;

}

public string verificaLogin(string loginwpf, string senhawpf)

{

Loginn log = new Loginn();

log.login = loginwpf;

log.senha = senhawpf;

var result = db.Loginns.Where(l => l.login == log.login && l.senha == log.senha).ToList();

if (result.Count() > 0)

{

if (result[0].idNivelAcesso == 1)

{

return "true";

}

return "false";

}

return "false";

}

}

}

**MODELS**

using System;

using System.ComponentModel.DataAnnotations;

using System.Web.Mvc;

namespace UniinfoAsp.Models

{

[MetadataType(typeof(FuncionarioMetadata))]

public partial class Funcionario

{

}

public partial class FuncionarioMetadata

{

[Required(ErrorMessage = "Campo vazio", AllowEmptyStrings = false)]

public int idFuncionario { get; set; }

[Required(ErrorMessage = "Campo vazio", AllowEmptyStrings = false)]

public string nome { get; set; }

[Required(ErrorMessage = "Campo vazio", AllowEmptyStrings = false)]

public Nullable<int> ramal { get; set; }

//[MaxLength(30)]

[Required(ErrorMessage = "Campo vazio", AllowEmptyStrings = false)]

//[Remote("computadorCadastrado", "Funcionario", ErrorMessage = "Esse computador ja está cadastrado")]

public Nullable<int> nComputador { get; set; }

[Required(ErrorMessage = "Campo vazio", AllowEmptyStrings = false)]

public string email { get; set; }

[Required(ErrorMessage = "Campo vazio", AllowEmptyStrings = false)]

public string setor { get; set; }

}

[MetadataType(typeof(LoginnMetadata))]

public partial class Loginn

{

}

public partial class LoginnMetadata

{

[Required(ErrorMessage = "Campo vazio", AllowEmptyStrings = false)]

public int idNivelAcesso { get; set; }

[Required(ErrorMessage = "Campo vazio", AllowEmptyStrings = false)]

[System.Web.Mvc.Remote("loginExistente", "Loginn", ErrorMessage = "Login existente")]

public string login { get; set; }

[Required(ErrorMessage = "Campo vazio", AllowEmptyStrings = false)]

[DataType(DataType.Password)]

public string senha { get; set; }

}

[MetadataType(typeof(ChamadoMetadata))]

public partial class Chamado

{

}

public partial class ChamadoMetadata

{

[Required(ErrorMessage = "Campo vazio", AllowEmptyStrings = false)]

public int idFuncionario { get; set; }

[Required(ErrorMessage = "Campo vazio", AllowEmptyStrings = false)]

public int idProblema { get; set; }

[Required(ErrorMessage = "Campo vazio", AllowEmptyStrings = false)]

public string descricao { get; set; }

[Required(ErrorMessage = "Campo vazio", AllowEmptyStrings = false)]

public System.DateTime dataChamado { get; set; }

[Required(ErrorMessage = "Campo vazio", AllowEmptyStrings = false)]

public string statusAtendimento { get; set; }

}

[MetadataType(typeof(ProblemaMetadata))]

public partial class Problema

{

}

public partial class ProblemaMetadata

{

[Remote("problemaExistente", "Problema", ErrorMessage = "Problema existente")]

[Required(ErrorMessage = "Campo vazio", AllowEmptyStrings = false)]

public string tipoProblema { get; set; }

}

}

using System.Linq;

using System.Web.Security;

namespace UniinfoAsp.Models

{

public class SiteRole : RoleProvider

{

public override string ApplicationName { get => throw new System.NotImplementedException(); set => throw new System.NotImplementedException(); }

public override void AddUsersToRoles(string[] usernames, string[] roleNames)

{

throw new System.NotImplementedException();

}

public override void CreateRole(string roleName)

{

throw new System.NotImplementedException();

}

public override bool DeleteRole(string roleName, bool throwOnPopulatedRole)

{

throw new System.NotImplementedException();

}

public override string[] FindUsersInRole(string roleName, string usernameToMatch)

{

throw new System.NotImplementedException();

}

public override string[] GetAllRoles()

{

throw new System.NotImplementedException();

}

public override string[] GetRolesForUser(string username)

{

UnipEntities db = new UnipEntities();

string date = db.Loginns.Where(l => l.login == username).SingleOrDefault().nivelAcesso.tipoAcesso;

string[] result = { date };

return result;

}

public override string[] GetUsersInRole(string roleName)

{

throw new System.NotImplementedException();

}

public override bool IsUserInRole(string username, string roleName)

{

throw new System.NotImplementedException();

}

public override void RemoveUsersFromRoles(string[] usernames, string[] roleNames)

{

throw new System.NotImplementedException();

}

public override bool RoleExists(string roleName)

{

throw new System.NotImplementedException();

}

}

}

<#@ template language="C#" debug="false" hostspecific="true"#>

<#@ include file="EF6.Utility.CS.ttinclude"#><#@

output extension=".cs"#><#

const string inputFile = @"Uniinfo.edmx";

var textTransform = DynamicTextTransformation.Create(this);

var code = new CodeGenerationTools(this);

var ef = new MetadataTools(this);

var typeMapper = new TypeMapper(code, ef, textTransform.Errors);

var loader = new EdmMetadataLoader(textTransform.Host, textTransform.Errors);

var itemCollection = loader.CreateEdmItemCollection(inputFile);

var modelNamespace = loader.GetModelNamespace(inputFile);

var codeStringGenerator = new CodeStringGenerator(code, typeMapper, ef);

var container = itemCollection.OfType<EntityContainer>().FirstOrDefault();

if (container == null)

{

return string.Empty;

}

#>

//------------------------------------------------------------------------------

// <auto-generated>

// <#=CodeGenerationTools.GetResourceString("Template\_GeneratedCodeCommentLine1")#>

//

// <#=CodeGenerationTools.GetResourceString("Template\_GeneratedCodeCommentLine2")#>

// <#=CodeGenerationTools.GetResourceString("Template\_GeneratedCodeCommentLine3")#>

// </auto-generated>

//------------------------------------------------------------------------------

<#

var codeNamespace = code.VsNamespaceSuggestion();

if (!String.IsNullOrEmpty(codeNamespace))

{

#>

namespace <#=code.EscapeNamespace(codeNamespace)#>

{

<#

PushIndent(" ");

}

#>

using System;

using System.Data.Entity;

using System.Data.Entity.Infrastructure;

<#

if (container.FunctionImports.Any())

{

#>

using System.Data.Entity.Core.Objects;

using System.Linq;

<#

}

#>

<#=Accessibility.ForType(container)#> partial class <#=code.Escape(container)#> : DbContext

{

public <#=code.Escape(container)#>()

: base("name=<#=container.Name#>")

{

<#

if (!loader.IsLazyLoadingEnabled(container))

{

#>

this.Configuration.LazyLoadingEnabled = false;

<#

}

foreach (var entitySet in container.BaseEntitySets.OfType<EntitySet>())

{

// Note: the DbSet members are defined below such that the getter and

// setter always have the same accessibility as the DbSet definition

if (Accessibility.ForReadOnlyProperty(entitySet) != "public")

{

#>

<#=codeStringGenerator.DbSetInitializer(entitySet)#>

<#

}

}

#>

}

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

throw new UnintentionalCodeFirstException();

}

<#

foreach (var entitySet in container.BaseEntitySets.OfType<EntitySet>())

{

#>

<#=codeStringGenerator.DbSet(entitySet)#>

<#

}

foreach (var edmFunction in container.FunctionImports)

{

WriteFunctionImport(typeMapper, codeStringGenerator, edmFunction, modelNamespace, includeMergeOption: false);

}

#>

}

<#

if (!String.IsNullOrEmpty(codeNamespace))

{

PopIndent();

#>

}

<#

}

#>

<#+

private void WriteFunctionImport(TypeMapper typeMapper, CodeStringGenerator codeStringGenerator, EdmFunction edmFunction, string modelNamespace, bool includeMergeOption)

{

if (typeMapper.IsComposable(edmFunction))

{

#>

[DbFunction("<#=edmFunction.NamespaceName#>", "<#=edmFunction.Name#>")]

<#=codeStringGenerator.ComposableFunctionMethod(edmFunction, modelNamespace)#>

{

<#+

codeStringGenerator.WriteFunctionParameters(edmFunction, WriteFunctionParameter);

#>

<#=codeStringGenerator.ComposableCreateQuery(edmFunction, modelNamespace)#>

}

<#+

}

else

{

#>

<#=codeStringGenerator.FunctionMethod(edmFunction, modelNamespace, includeMergeOption)#>

{

<#+

codeStringGenerator.WriteFunctionParameters(edmFunction, WriteFunctionParameter);

#>

<#=codeStringGenerator.ExecuteFunction(edmFunction, modelNamespace, includeMergeOption)#>

}

<#+

if (typeMapper.GenerateMergeOptionFunction(edmFunction, includeMergeOption))

{

WriteFunctionImport(typeMapper, codeStringGenerator, edmFunction, modelNamespace, includeMergeOption: true);

}

}

}

public void WriteFunctionParameter(string name, string isNotNull, string notNullInit, string nullInit)

{

#>

var <#=name#> = <#=isNotNull#> ?

<#=notNullInit#> :

<#=nullInit#>;

<#+

}

public const string TemplateId = "CSharp\_DbContext\_Context\_EF6";

public class CodeStringGenerator

{

private readonly CodeGenerationTools \_code;

private readonly TypeMapper \_typeMapper;

private readonly MetadataTools \_ef;

public CodeStringGenerator(CodeGenerationTools code, TypeMapper typeMapper, MetadataTools ef)

{

ArgumentNotNull(code, "code");

ArgumentNotNull(typeMapper, "typeMapper");

ArgumentNotNull(ef, "ef");

\_code = code;

\_typeMapper = typeMapper;

\_ef = ef;

}

public string Property(EdmProperty edmProperty)

{

return string.Format(

CultureInfo.InvariantCulture,

"{0} {1} {2} {{ {3}get; {4}set; }}",

Accessibility.ForProperty(edmProperty),

\_typeMapper.GetTypeName(edmProperty.TypeUsage),

\_code.Escape(edmProperty),

\_code.SpaceAfter(Accessibility.ForGetter(edmProperty)),

\_code.SpaceAfter(Accessibility.ForSetter(edmProperty)));

}

public string NavigationProperty(NavigationProperty navProp)

{

var endType = \_typeMapper.GetTypeName(navProp.ToEndMember.GetEntityType());

return string.Format(

CultureInfo.InvariantCulture,

"{0} {1} {2} {{ {3}get; {4}set; }}",

AccessibilityAndVirtual(Accessibility.ForNavigationProperty(navProp)),

navProp.ToEndMember.RelationshipMultiplicity == RelationshipMultiplicity.Many ? ("ICollection<" + endType + ">") : endType,

\_code.Escape(navProp),

\_code.SpaceAfter(Accessibility.ForGetter(navProp)),

\_code.SpaceAfter(Accessibility.ForSetter(navProp)));

}

public string AccessibilityAndVirtual(string accessibility)

{

return accessibility + (accessibility != "private" ? " virtual" : "");

}

public string EntityClassOpening(EntityType entity)

{

return string.Format(

CultureInfo.InvariantCulture,

"{0} {1}partial class {2}{3}",

Accessibility.ForType(entity),

\_code.SpaceAfter(\_code.AbstractOption(entity)),

\_code.Escape(entity),

\_code.StringBefore(" : ", \_typeMapper.GetTypeName(entity.BaseType)));

}

public string EnumOpening(SimpleType enumType)

{

return string.Format(

CultureInfo.InvariantCulture,

"{0} enum {1} : {2}",

Accessibility.ForType(enumType),

\_code.Escape(enumType),

\_code.Escape(\_typeMapper.UnderlyingClrType(enumType)));

}

public void WriteFunctionParameters(EdmFunction edmFunction, Action<string, string, string, string> writeParameter)

{

var parameters = FunctionImportParameter.Create(edmFunction.Parameters, \_code, \_ef);

foreach (var parameter in parameters.Where(p => p.NeedsLocalVariable))

{

var isNotNull = parameter.IsNullableOfT ? parameter.FunctionParameterName + ".HasValue" : parameter.FunctionParameterName + " != null";

var notNullInit = "new ObjectParameter(\"" + parameter.EsqlParameterName + "\", " + parameter.FunctionParameterName + ")";

var nullInit = "new ObjectParameter(\"" + parameter.EsqlParameterName + "\", typeof(" + TypeMapper.FixNamespaces(parameter.RawClrTypeName) + "))";

writeParameter(parameter.LocalVariableName, isNotNull, notNullInit, nullInit);

}

}

public string ComposableFunctionMethod(EdmFunction edmFunction, string modelNamespace)

{

var parameters = \_typeMapper.GetParameters(edmFunction);

return string.Format(

CultureInfo.InvariantCulture,

"{0} IQueryable<{1}> {2}({3})",

AccessibilityAndVirtual(Accessibility.ForMethod(edmFunction)),

\_typeMapper.GetTypeName(\_typeMapper.GetReturnType(edmFunction), modelNamespace),

\_code.Escape(edmFunction),

string.Join(", ", parameters.Select(p => TypeMapper.FixNamespaces(p.FunctionParameterType) + " " + p.FunctionParameterName).ToArray()));

}

public string ComposableCreateQuery(EdmFunction edmFunction, string modelNamespace)

{

var parameters = \_typeMapper.GetParameters(edmFunction);

return string.Format(

CultureInfo.InvariantCulture,

"return ((IObjectContextAdapter)this).ObjectContext.CreateQuery<{0}>(\"[{1}].[{2}]({3})\"{4});",

\_typeMapper.GetTypeName(\_typeMapper.GetReturnType(edmFunction), modelNamespace),

edmFunction.NamespaceName,

edmFunction.Name,

string.Join(", ", parameters.Select(p => "@" + p.EsqlParameterName).ToArray()),

\_code.StringBefore(", ", string.Join(", ", parameters.Select(p => p.ExecuteParameterName).ToArray())));

}

public string FunctionMethod(EdmFunction edmFunction, string modelNamespace, bool includeMergeOption)

{

var parameters = \_typeMapper.GetParameters(edmFunction);

var returnType = \_typeMapper.GetReturnType(edmFunction);

var paramList = String.Join(", ", parameters.Select(p => TypeMapper.FixNamespaces(p.FunctionParameterType) + " " + p.FunctionParameterName).ToArray());

if (includeMergeOption)

{

paramList = \_code.StringAfter(paramList, ", ") + "MergeOption mergeOption";

}

return string.Format(

CultureInfo.InvariantCulture,

"{0} {1} {2}({3})",

AccessibilityAndVirtual(Accessibility.ForMethod(edmFunction)),

returnType == null ? "int" : "ObjectResult<" + \_typeMapper.GetTypeName(returnType, modelNamespace) + ">",

\_code.Escape(edmFunction),

paramList);

}

public string ExecuteFunction(EdmFunction edmFunction, string modelNamespace, bool includeMergeOption)

{

var parameters = \_typeMapper.GetParameters(edmFunction);

var returnType = \_typeMapper.GetReturnType(edmFunction);

var callParams = \_code.StringBefore(", ", String.Join(", ", parameters.Select(p => p.ExecuteParameterName).ToArray()));

if (includeMergeOption)

{

callParams = ", mergeOption" + callParams;

}

return string.Format(

CultureInfo.InvariantCulture,

"return ((IObjectContextAdapter)this).ObjectContext.ExecuteFunction{0}(\"{1}\"{2});",

returnType == null ? "" : "<" + \_typeMapper.GetTypeName(returnType, modelNamespace) + ">",

edmFunction.Name,

callParams);

}

public string DbSet(EntitySet entitySet)

{

return string.Format(

CultureInfo.InvariantCulture,

"{0} virtual DbSet<{1}> {2} {{ get; set; }}",

Accessibility.ForReadOnlyProperty(entitySet),

\_typeMapper.GetTypeName(entitySet.ElementType),

\_code.Escape(entitySet));

}

public string DbSetInitializer(EntitySet entitySet)

{

return string.Format(

CultureInfo.InvariantCulture,

"{0} = Set<{1}>();",

\_code.Escape(entitySet),

\_typeMapper.GetTypeName(entitySet.ElementType));

}

public string UsingDirectives(bool inHeader, bool includeCollections = true)

{

return inHeader == string.IsNullOrEmpty(\_code.VsNamespaceSuggestion())

? string.Format(

CultureInfo.InvariantCulture,

"{0}using System;{1}" +

"{2}",

inHeader ? Environment.NewLine : "",

includeCollections ? (Environment.NewLine + "using System.Collections.Generic;") : "",

inHeader ? "" : Environment.NewLine)

: "";

}

}

public class TypeMapper

{

private const string ExternalTypeNameAttributeName = @"http://schemas.microsoft.com/ado/2006/04/codegeneration:ExternalTypeName";

private readonly System.Collections.IList \_errors;

private readonly CodeGenerationTools \_code;

private readonly MetadataTools \_ef;

public static string FixNamespaces(string typeName)

{

return typeName.Replace("System.Data.Spatial.", "System.Data.Entity.Spatial.");

}

public TypeMapper(CodeGenerationTools code, MetadataTools ef, System.Collections.IList errors)

{

ArgumentNotNull(code, "code");

ArgumentNotNull(ef, "ef");

ArgumentNotNull(errors, "errors");

\_code = code;

\_ef = ef;

\_errors = errors;

}

public string GetTypeName(TypeUsage typeUsage)

{

return typeUsage == null ? null : GetTypeName(typeUsage.EdmType, \_ef.IsNullable(typeUsage), modelNamespace: null);

}

public string GetTypeName(EdmType edmType)

{

return GetTypeName(edmType, isNullable: null, modelNamespace: null);

}

public string GetTypeName(TypeUsage typeUsage, string modelNamespace)

{

return typeUsage == null ? null : GetTypeName(typeUsage.EdmType, \_ef.IsNullable(typeUsage), modelNamespace);

}

public string GetTypeName(EdmType edmType, string modelNamespace)

{

return GetTypeName(edmType, isNullable: null, modelNamespace: modelNamespace);

}

public string GetTypeName(EdmType edmType, bool? isNullable, string modelNamespace)

{

if (edmType == null)

{

return null;

}

var collectionType = edmType as CollectionType;

if (collectionType != null)

{

return String.Format(CultureInfo.InvariantCulture, "ICollection<{0}>", GetTypeName(collectionType.TypeUsage, modelNamespace));

}

var typeName = \_code.Escape(edmType.MetadataProperties

.Where(p => p.Name == ExternalTypeNameAttributeName)

.Select(p => (string)p.Value)

.FirstOrDefault())

?? (modelNamespace != null && edmType.NamespaceName != modelNamespace ?

\_code.CreateFullName(\_code.EscapeNamespace(edmType.NamespaceName), \_code.Escape(edmType)) :

\_code.Escape(edmType));

if (edmType is StructuralType)

{

return typeName;

}

if (edmType is SimpleType)

{

var clrType = UnderlyingClrType(edmType);

if (!IsEnumType(edmType))

{

typeName = \_code.Escape(clrType);

}

typeName = FixNamespaces(typeName);

return clrType.IsValueType && isNullable == true ?

String.Format(CultureInfo.InvariantCulture, "Nullable<{0}>", typeName) :

typeName;

}

throw new ArgumentException("edmType");

}

public Type UnderlyingClrType(EdmType edmType)

{

ArgumentNotNull(edmType, "edmType");

var primitiveType = edmType as PrimitiveType;

if (primitiveType != null)

{

return primitiveType.ClrEquivalentType;

}

if (IsEnumType(edmType))

{

return GetEnumUnderlyingType(edmType).ClrEquivalentType;

}

return typeof(object);

}

public object GetEnumMemberValue(MetadataItem enumMember)

{

ArgumentNotNull(enumMember, "enumMember");

var valueProperty = enumMember.GetType().GetProperty("Value");

return valueProperty == null ? null : valueProperty.GetValue(enumMember, null);

}

public string GetEnumMemberName(MetadataItem enumMember)

{

ArgumentNotNull(enumMember, "enumMember");

var nameProperty = enumMember.GetType().GetProperty("Name");

return nameProperty == null ? null : (string)nameProperty.GetValue(enumMember, null);

}

public System.Collections.IEnumerable GetEnumMembers(EdmType enumType)

{

ArgumentNotNull(enumType, "enumType");

var membersProperty = enumType.GetType().GetProperty("Members");

return membersProperty != null

? (System.Collections.IEnumerable)membersProperty.GetValue(enumType, null)

: Enumerable.Empty<MetadataItem>();

}

public bool EnumIsFlags(EdmType enumType)

{

ArgumentNotNull(enumType, "enumType");

var isFlagsProperty = enumType.GetType().GetProperty("IsFlags");

return isFlagsProperty != null && (bool)isFlagsProperty.GetValue(enumType, null);

}

public bool IsEnumType(GlobalItem edmType)

{

ArgumentNotNull(edmType, "edmType");

return edmType.GetType().Name == "EnumType";

}

public PrimitiveType GetEnumUnderlyingType(EdmType enumType)

{

ArgumentNotNull(enumType, "enumType");

return (PrimitiveType)enumType.GetType().GetProperty("UnderlyingType").GetValue(enumType, null);

}

public string CreateLiteral(object value)

{

if (value == null || value.GetType() != typeof(TimeSpan))

{

return \_code.CreateLiteral(value);

}

return string.Format(CultureInfo.InvariantCulture, "new TimeSpan({0})", ((TimeSpan)value).Ticks);

}

public bool VerifyCaseInsensitiveTypeUniqueness(IEnumerable<string> types, string sourceFile)

{

ArgumentNotNull(types, "types");

ArgumentNotNull(sourceFile, "sourceFile");

var hash = new HashSet<string>(StringComparer.InvariantCultureIgnoreCase);

if (types.Any(item => !hash.Add(item)))

{

\_errors.Add(

new CompilerError(sourceFile, -1, -1, "6023",

String.Format(CultureInfo.CurrentCulture, CodeGenerationTools.GetResourceString("Template\_CaseInsensitiveTypeConflict"))));

return false;

}

return true;

}

public IEnumerable<SimpleType> GetEnumItemsToGenerate(IEnumerable<GlobalItem> itemCollection)

{

return GetItemsToGenerate<SimpleType>(itemCollection)

.Where(e => IsEnumType(e));

}

public IEnumerable<T> GetItemsToGenerate<T>(IEnumerable<GlobalItem> itemCollection) where T: EdmType

{

return itemCollection

.OfType<T>()

.Where(i => !i.MetadataProperties.Any(p => p.Name == ExternalTypeNameAttributeName))

.OrderBy(i => i.Name);

}

public IEnumerable<string> GetAllGlobalItems(IEnumerable<GlobalItem> itemCollection)

{

return itemCollection

.Where(i => i is EntityType || i is ComplexType || i is EntityContainer || IsEnumType(i))

.Select(g => GetGlobalItemName(g));

}

public string GetGlobalItemName(GlobalItem item)

{

if (item is EdmType)

{

return ((EdmType)item).Name;

}

else

{

return ((EntityContainer)item).Name;

}

}

public IEnumerable<EdmProperty> GetSimpleProperties(EntityType type)

{

return type.Properties.Where(p => p.TypeUsage.EdmType is SimpleType && p.DeclaringType == type);

}

public IEnumerable<EdmProperty> GetSimpleProperties(ComplexType type)

{

return type.Properties.Where(p => p.TypeUsage.EdmType is SimpleType && p.DeclaringType == type);

}

public IEnumerable<EdmProperty> GetComplexProperties(EntityType type)

{

return type.Properties.Where(p => p.TypeUsage.EdmType is ComplexType && p.DeclaringType == type);

}

public IEnumerable<EdmProperty> GetComplexProperties(ComplexType type)

{

return type.Properties.Where(p => p.TypeUsage.EdmType is ComplexType && p.DeclaringType == type);

}

public IEnumerable<EdmProperty> GetPropertiesWithDefaultValues(EntityType type)

{

return type.Properties.Where(p => p.TypeUsage.EdmType is SimpleType && p.DeclaringType == type && p.DefaultValue != null);

}

public IEnumerable<EdmProperty> GetPropertiesWithDefaultValues(ComplexType type)

{

return type.Properties.Where(p => p.TypeUsage.EdmType is SimpleType && p.DeclaringType == type && p.DefaultValue != null);

}

public IEnumerable<NavigationProperty> GetNavigationProperties(EntityType type)

{

return type.NavigationProperties.Where(np => np.DeclaringType == type);

}

public IEnumerable<NavigationProperty> GetCollectionNavigationProperties(EntityType type)

{

return type.NavigationProperties.Where(np => np.DeclaringType == type && np.ToEndMember.RelationshipMultiplicity == RelationshipMultiplicity.Many);

}

public FunctionParameter GetReturnParameter(EdmFunction edmFunction)

{

ArgumentNotNull(edmFunction, "edmFunction");

var returnParamsProperty = edmFunction.GetType().GetProperty("ReturnParameters");

return returnParamsProperty == null

? edmFunction.ReturnParameter

: ((IEnumerable<FunctionParameter>)returnParamsProperty.GetValue(edmFunction, null)).FirstOrDefault();

}

public bool IsComposable(EdmFunction edmFunction)

{

ArgumentNotNull(edmFunction, "edmFunction");

var isComposableProperty = edmFunction.GetType().GetProperty("IsComposableAttribute");

return isComposableProperty != null && (bool)isComposableProperty.GetValue(edmFunction, null);

}

public IEnumerable<FunctionImportParameter> GetParameters(EdmFunction edmFunction)

{

return FunctionImportParameter.Create(edmFunction.Parameters, \_code, \_ef);

}

public TypeUsage GetReturnType(EdmFunction edmFunction)

{

var returnParam = GetReturnParameter(edmFunction);

return returnParam == null ? null : \_ef.GetElementType(returnParam.TypeUsage);

}

public bool GenerateMergeOptionFunction(EdmFunction edmFunction, bool includeMergeOption)

{

var returnType = GetReturnType(edmFunction);

return !includeMergeOption && returnType != null && returnType.EdmType.BuiltInTypeKind == BuiltInTypeKind.EntityType;

}

}

public static void ArgumentNotNull<T>(T arg, string name) where T : class

{

if (arg == null)

{

throw new ArgumentNullException(name);

}

}

#>

//------------------------------------------------------------------------------

// <auto-generated>

// This code was generated from a template.

//

// Manual changes to this file may cause unexpected behavior in your application.

// Manual changes to this file will be overwritten if the code is regenerated.

// </auto-generated>

//------------------------------------------------------------------------------

namespace UniinfoAsp.Models

{

using System;

using System.Data.Entity;

using System.Data.Entity.Infrastructure;

public partial class UnipEntities : DbContext

{

public UnipEntities()

: base("UnipEntities")

{

}

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

throw new UnintentionalCodeFirstException();

}

public virtual DbSet<Chamado> Chamadoes { get; set; }

public virtual DbSet<chamadoAtendimento> chamadoAtendimentoes { get; set; }

public virtual DbSet<Funcionario> Funcionarios { get; set; }

public virtual DbSet<Loginn> Loginns { get; set; }

public virtual DbSet<nivelAcesso> nivelAcessoes { get; set; }

public virtual DbSet<Problema> Problemas { get; set; }

public virtual DbSet<Con\_Chamado> Con\_Chamado { get; set; }

}

}

// T4 code generation is enabled for model 'C:\Users\OZ\Desktop\gitpim\pim\_2018\_2\UniinfoAsp\UniinfoAsp\Models\Uniinfo.edmx'.

// To enable legacy code generation, change the value of the 'Code Generation Strategy' designer

// property to 'Legacy ObjectContext'. This property is available in the Properties Window when the model

// is open in the designer.

// If no context and entity classes have been generated, it may be because you created an empty model but

// have not yet chosen which version of Entity Framework to use. To generate a context class and entity

// classes for your model, open the model in the designer, right-click on the designer surface, and

// select 'Update Model from Database...', 'Generate Database from Model...', or 'Add Code Generation

// Item...'.

<?xml version="1.0" encoding="utf-8"?>

<edmx:Edmx Version="3.0" xmlns:edmx="http://schemas.microsoft.com/ado/2009/11/edmx">

<!-- EF Designer content (DO NOT EDIT MANUALLY BELOW HERE) -->

<edmx:Designer xmlns="http://schemas.microsoft.com/ado/2009/11/edmx">

<!-- Diagram content (shape and connector positions) -->

<edmx:Diagrams>

<Diagram DiagramId="a23284fd252c416e92619ffc0b105e54" Name="Diagram1">

<EntityTypeShape EntityType="UnipModelEntities.Chamado" Width="1.5" PointX="3" PointY="3.5" IsExpanded="true" />

<EntityTypeShape EntityType="UnipModelEntities.chamadoAtendimento" Width="1.5" PointX="5.25" PointY="0.75" IsExpanded="true" />

<EntityTypeShape EntityType="UnipModelEntities.Funcionario" Width="1.5" PointX="0.75" PointY="3.5" IsExpanded="true" />

<EntityTypeShape EntityType="UnipModelEntities.Loginn" Width="1.5" PointX="6" PointY="3.625" IsExpanded="true" />

<EntityTypeShape EntityType="UnipModelEntities.nivelAcesso" Width="1.5" PointX="3.75" PointY="7" IsExpanded="true" />

<EntityTypeShape EntityType="UnipModelEntities.Problema" Width="1.5" PointX="0.75" PointY="1" IsExpanded="true" />

<EntityTypeShape EntityType="UnipModelEntities.Con\_Chamado" Width="1.5" PointX="0.75" PointY="7.5" IsExpanded="true" />

<AssociationConnector Association="UnipModelEntities.FK\_\_Chamado\_\_idFunci\_\_5441852A" ManuallyRouted="false" />

<AssociationConnector Association="UnipModelEntities.FK\_\_Chamado\_\_idProbl\_\_5535A963" ManuallyRouted="false" />

<AssociationConnector Association="UnipModelEntities.FK\_\_chamadoAt\_\_idCha\_\_5629CD9C" ManuallyRouted="false" />

<AssociationConnector Association="UnipModelEntities.FK\_\_chamadoAt\_\_idFun\_\_571DF1D5" ManuallyRouted="false" />

<AssociationConnector Association="UnipModelEntities.FK\_\_Loginn\_\_idFuncio\_\_5812160E" ManuallyRouted="false" />

<AssociationConnector Association="UnipModelEntities.FK\_\_Loginn\_\_idNivelA\_\_59063A47" ManuallyRouted="false" />

</Diagram>

</edmx:Diagrams>

</edmx:Designer>

</edmx:Edmx>

<#@ template language="C#" debug="false" hostspecific="true"#>

<#@ include file="EF6.Utility.CS.ttinclude"#><#@

output extension=".cs"#><#

const string inputFile = @"Uniinfo.edmx";

var textTransform = DynamicTextTransformation.Create(this);

var code = new CodeGenerationTools(this);

var ef = new MetadataTools(this);

var typeMapper = new TypeMapper(code, ef, textTransform.Errors);

var fileManager = EntityFrameworkTemplateFileManager.Create(this);

var itemCollection = new EdmMetadataLoader(textTransform.Host, textTransform.Errors).CreateEdmItemCollection(inputFile);

var codeStringGenerator = new CodeStringGenerator(code, typeMapper, ef);

if (!typeMapper.VerifyCaseInsensitiveTypeUniqueness(typeMapper.GetAllGlobalItems(itemCollection), inputFile))

{

return string.Empty;

}

WriteHeader(codeStringGenerator, fileManager);

foreach (var entity in typeMapper.GetItemsToGenerate<EntityType>(itemCollection))

{

fileManager.StartNewFile(entity.Name + ".cs");

BeginNamespace(code);

#>

<#=codeStringGenerator.UsingDirectives(inHeader: false)#>

<#=codeStringGenerator.EntityClassOpening(entity)#>

{

<#

var propertiesWithDefaultValues = typeMapper.GetPropertiesWithDefaultValues(entity);

var collectionNavigationProperties = typeMapper.GetCollectionNavigationProperties(entity);

var complexProperties = typeMapper.GetComplexProperties(entity);

if (propertiesWithDefaultValues.Any() || collectionNavigationProperties.Any() || complexProperties.Any())

{

#>

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]

public <#=code.Escape(entity)#>()

{

<#

foreach (var edmProperty in propertiesWithDefaultValues)

{

#>

this.<#=code.Escape(edmProperty)#> = <#=typeMapper.CreateLiteral(edmProperty.DefaultValue)#>;

<#

}

foreach (var navigationProperty in collectionNavigationProperties)

{

#>

this.<#=code.Escape(navigationProperty)#> = new HashSet<<#=typeMapper.GetTypeName(navigationProperty.ToEndMember.GetEntityType())#>>();

<#

}

foreach (var complexProperty in complexProperties)

{

#>

this.<#=code.Escape(complexProperty)#> = new <#=typeMapper.GetTypeName(complexProperty.TypeUsage)#>();

<#

}

#>

}

<#

}

var simpleProperties = typeMapper.GetSimpleProperties(entity);

if (simpleProperties.Any())

{

foreach (var edmProperty in simpleProperties)

{

#>

<#=codeStringGenerator.Property(edmProperty)#>

<#

}

}

if (complexProperties.Any())

{

#>

<#

foreach(var complexProperty in complexProperties)

{

#>

<#=codeStringGenerator.Property(complexProperty)#>

<#

}

}

var navigationProperties = typeMapper.GetNavigationProperties(entity);

if (navigationProperties.Any())

{

#>

<#

foreach (var navigationProperty in navigationProperties)

{

if (navigationProperty.ToEndMember.RelationshipMultiplicity == RelationshipMultiplicity.Many)

{

#>

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]

<#

}

#>

<#=codeStringGenerator.NavigationProperty(navigationProperty)#>

<#

}

}

#>

}

<#

EndNamespace(code);

}

foreach (var complex in typeMapper.GetItemsToGenerate<ComplexType>(itemCollection))

{

fileManager.StartNewFile(complex.Name + ".cs");

BeginNamespace(code);

#>

<#=codeStringGenerator.UsingDirectives(inHeader: false, includeCollections: false)#>

<#=Accessibility.ForType(complex)#> partial class <#=code.Escape(complex)#>

{

<#

var complexProperties = typeMapper.GetComplexProperties(complex);

var propertiesWithDefaultValues = typeMapper.GetPropertiesWithDefaultValues(complex);

if (propertiesWithDefaultValues.Any() || complexProperties.Any())

{

#>

public <#=code.Escape(complex)#>()

{

<#

foreach (var edmProperty in propertiesWithDefaultValues)

{

#>

this.<#=code.Escape(edmProperty)#> = <#=typeMapper.CreateLiteral(edmProperty.DefaultValue)#>;

<#

}

foreach (var complexProperty in complexProperties)

{

#>

this.<#=code.Escape(complexProperty)#> = new <#=typeMapper.GetTypeName(complexProperty.TypeUsage)#>();

<#

}

#>

}

<#

}

var simpleProperties = typeMapper.GetSimpleProperties(complex);

if (simpleProperties.Any())

{

foreach(var edmProperty in simpleProperties)

{

#>

<#=codeStringGenerator.Property(edmProperty)#>

<#

}

}

if (complexProperties.Any())

{

#>

<#

foreach(var edmProperty in complexProperties)

{

#>

<#=codeStringGenerator.Property(edmProperty)#>

<#

}

}

#>

}

<#

EndNamespace(code);

}

foreach (var enumType in typeMapper.GetEnumItemsToGenerate(itemCollection))

{

fileManager.StartNewFile(enumType.Name + ".cs");

BeginNamespace(code);

#>

<#=codeStringGenerator.UsingDirectives(inHeader: false, includeCollections: false)#>

<#

if (typeMapper.EnumIsFlags(enumType))

{

#>

[Flags]

<#

}

#>

<#=codeStringGenerator.EnumOpening(enumType)#>

{

<#

var foundOne = false;

foreach (MetadataItem member in typeMapper.GetEnumMembers(enumType))

{

foundOne = true;

#>

<#=code.Escape(typeMapper.GetEnumMemberName(member))#> = <#=typeMapper.GetEnumMemberValue(member)#>,

<#

}

if (foundOne)

{

this.GenerationEnvironment.Remove(this.GenerationEnvironment.Length - 3, 1);

}

#>

}

<#

EndNamespace(code);

}

fileManager.Process();

#>

<#+

public void WriteHeader(CodeStringGenerator codeStringGenerator, EntityFrameworkTemplateFileManager fileManager)

{

fileManager.StartHeader();

#>

//------------------------------------------------------------------------------

// <auto-generated>

// <#=CodeGenerationTools.GetResourceString("Template\_GeneratedCodeCommentLine1")#>

//

// <#=CodeGenerationTools.GetResourceString("Template\_GeneratedCodeCommentLine2")#>

// <#=CodeGenerationTools.GetResourceString("Template\_GeneratedCodeCommentLine3")#>

// </auto-generated>

//------------------------------------------------------------------------------

<#=codeStringGenerator.UsingDirectives(inHeader: true)#>

<#+

fileManager.EndBlock();

}

public void BeginNamespace(CodeGenerationTools code)

{

var codeNamespace = code.VsNamespaceSuggestion();

if (!String.IsNullOrEmpty(codeNamespace))

{

#>

namespace <#=code.EscapeNamespace(codeNamespace)#>

{

<#+

PushIndent(" ");

}

}

public void EndNamespace(CodeGenerationTools code)

{

if (!String.IsNullOrEmpty(code.VsNamespaceSuggestion()))

{

PopIndent();

#>

}

<#+

}

}

public const string TemplateId = "CSharp\_DbContext\_Types\_EF6";

public class CodeStringGenerator

{

private readonly CodeGenerationTools \_code;

private readonly TypeMapper \_typeMapper;

private readonly MetadataTools \_ef;

public CodeStringGenerator(CodeGenerationTools code, TypeMapper typeMapper, MetadataTools ef)

{

ArgumentNotNull(code, "code");

ArgumentNotNull(typeMapper, "typeMapper");

ArgumentNotNull(ef, "ef");

\_code = code;

\_typeMapper = typeMapper;

\_ef = ef;

}

public string Property(EdmProperty edmProperty)

{

return string.Format(

CultureInfo.InvariantCulture,

"{0} {1} {2} {{ {3}get; {4}set; }}",

Accessibility.ForProperty(edmProperty),

\_typeMapper.GetTypeName(edmProperty.TypeUsage),

\_code.Escape(edmProperty),

\_code.SpaceAfter(Accessibility.ForGetter(edmProperty)),

\_code.SpaceAfter(Accessibility.ForSetter(edmProperty)));

}

public string NavigationProperty(NavigationProperty navProp)

{

var endType = \_typeMapper.GetTypeName(navProp.ToEndMember.GetEntityType());

return string.Format(

CultureInfo.InvariantCulture,

"{0} {1} {2} {{ {3}get; {4}set; }}",

AccessibilityAndVirtual(Accessibility.ForNavigationProperty(navProp)),

navProp.ToEndMember.RelationshipMultiplicity == RelationshipMultiplicity.Many ? ("ICollection<" + endType + ">") : endType,

\_code.Escape(navProp),

\_code.SpaceAfter(Accessibility.ForGetter(navProp)),

\_code.SpaceAfter(Accessibility.ForSetter(navProp)));

}

public string AccessibilityAndVirtual(string accessibility)

{

return accessibility + (accessibility != "private" ? " virtual" : "");

}

public string EntityClassOpening(EntityType entity)

{

return string.Format(

CultureInfo.InvariantCulture,

"{0} {1}partial class {2}{3}",

Accessibility.ForType(entity),

\_code.SpaceAfter(\_code.AbstractOption(entity)),

\_code.Escape(entity),

\_code.StringBefore(" : ", \_typeMapper.GetTypeName(entity.BaseType)));

}

public string EnumOpening(SimpleType enumType)

{

return string.Format(

CultureInfo.InvariantCulture,

"{0} enum {1} : {2}",

Accessibility.ForType(enumType),

\_code.Escape(enumType),

\_code.Escape(\_typeMapper.UnderlyingClrType(enumType)));

}

public void WriteFunctionParameters(EdmFunction edmFunction, Action<string, string, string, string> writeParameter)

{

var parameters = FunctionImportParameter.Create(edmFunction.Parameters, \_code, \_ef);

foreach (var parameter in parameters.Where(p => p.NeedsLocalVariable))

{

var isNotNull = parameter.IsNullableOfT ? parameter.FunctionParameterName + ".HasValue" : parameter.FunctionParameterName + " != null";

var notNullInit = "new ObjectParameter(\"" + parameter.EsqlParameterName + "\", " + parameter.FunctionParameterName + ")";

var nullInit = "new ObjectParameter(\"" + parameter.EsqlParameterName + "\", typeof(" + TypeMapper.FixNamespaces(parameter.RawClrTypeName) + "))";

writeParameter(parameter.LocalVariableName, isNotNull, notNullInit, nullInit);

}

}

public string ComposableFunctionMethod(EdmFunction edmFunction, string modelNamespace)

{

var parameters = \_typeMapper.GetParameters(edmFunction);

return string.Format(

CultureInfo.InvariantCulture,

"{0} IQueryable<{1}> {2}({3})",

AccessibilityAndVirtual(Accessibility.ForMethod(edmFunction)),

\_typeMapper.GetTypeName(\_typeMapper.GetReturnType(edmFunction), modelNamespace),

\_code.Escape(edmFunction),

string.Join(", ", parameters.Select(p => TypeMapper.FixNamespaces(p.FunctionParameterType) + " " + p.FunctionParameterName).ToArray()));

}

public string ComposableCreateQuery(EdmFunction edmFunction, string modelNamespace)

{

var parameters = \_typeMapper.GetParameters(edmFunction);

return string.Format(

CultureInfo.InvariantCulture,

"return ((IObjectContextAdapter)this).ObjectContext.CreateQuery<{0}>(\"[{1}].[{2}]({3})\"{4});",

\_typeMapper.GetTypeName(\_typeMapper.GetReturnType(edmFunction), modelNamespace),

edmFunction.NamespaceName,

edmFunction.Name,

string.Join(", ", parameters.Select(p => "@" + p.EsqlParameterName).ToArray()),

\_code.StringBefore(", ", string.Join(", ", parameters.Select(p => p.ExecuteParameterName).ToArray())));

}

public string FunctionMethod(EdmFunction edmFunction, string modelNamespace, bool includeMergeOption)

{

var parameters = \_typeMapper.GetParameters(edmFunction);

var returnType = \_typeMapper.GetReturnType(edmFunction);

var paramList = String.Join(", ", parameters.Select(p => TypeMapper.FixNamespaces(p.FunctionParameterType) + " " + p.FunctionParameterName).ToArray());

if (includeMergeOption)

{

paramList = \_code.StringAfter(paramList, ", ") + "MergeOption mergeOption";

}

return string.Format(

CultureInfo.InvariantCulture,

"{0} {1} {2}({3})",

AccessibilityAndVirtual(Accessibility.ForMethod(edmFunction)),

returnType == null ? "int" : "ObjectResult<" + \_typeMapper.GetTypeName(returnType, modelNamespace) + ">",

\_code.Escape(edmFunction),

paramList);

}

public string ExecuteFunction(EdmFunction edmFunction, string modelNamespace, bool includeMergeOption)

{

var parameters = \_typeMapper.GetParameters(edmFunction);

var returnType = \_typeMapper.GetReturnType(edmFunction);

var callParams = \_code.StringBefore(", ", String.Join(", ", parameters.Select(p => p.ExecuteParameterName).ToArray()));

if (includeMergeOption)

{

callParams = ", mergeOption" + callParams;

}

return string.Format(

CultureInfo.InvariantCulture,

"return ((IObjectContextAdapter)this).ObjectContext.ExecuteFunction{0}(\"{1}\"{2});",

returnType == null ? "" : "<" + \_typeMapper.GetTypeName(returnType, modelNamespace) + ">",

edmFunction.Name,

callParams);

}

public string DbSet(EntitySet entitySet)

{

return string.Format(

CultureInfo.InvariantCulture,

"{0} virtual DbSet<{1}> {2} {{ get; set; }}",

Accessibility.ForReadOnlyProperty(entitySet),

\_typeMapper.GetTypeName(entitySet.ElementType),

\_code.Escape(entitySet));

}

public string UsingDirectives(bool inHeader, bool includeCollections = true)

{

return inHeader == string.IsNullOrEmpty(\_code.VsNamespaceSuggestion())

? string.Format(

CultureInfo.InvariantCulture,

"{0}using System;{1}" +

"{2}",

inHeader ? Environment.NewLine : "",

includeCollections ? (Environment.NewLine + "using System.Collections.Generic;") : "",

inHeader ? "" : Environment.NewLine)

: "";

}

}

public class TypeMapper

{

private const string ExternalTypeNameAttributeName = @"http://schemas.microsoft.com/ado/2006/04/codegeneration:ExternalTypeName";

private readonly System.Collections.IList \_errors;

private readonly CodeGenerationTools \_code;

private readonly MetadataTools \_ef;

public TypeMapper(CodeGenerationTools code, MetadataTools ef, System.Collections.IList errors)

{

ArgumentNotNull(code, "code");

ArgumentNotNull(ef, "ef");

ArgumentNotNull(errors, "errors");

\_code = code;

\_ef = ef;

\_errors = errors;

}

public static string FixNamespaces(string typeName)

{

return typeName.Replace("System.Data.Spatial.", "System.Data.Entity.Spatial.");

}

public string GetTypeName(TypeUsage typeUsage)

{

return typeUsage == null ? null : GetTypeName(typeUsage.EdmType, \_ef.IsNullable(typeUsage), modelNamespace: null);

}

public string GetTypeName(EdmType edmType)

{

return GetTypeName(edmType, isNullable: null, modelNamespace: null);

}

public string GetTypeName(TypeUsage typeUsage, string modelNamespace)

{

return typeUsage == null ? null : GetTypeName(typeUsage.EdmType, \_ef.IsNullable(typeUsage), modelNamespace);

}

public string GetTypeName(EdmType edmType, string modelNamespace)

{

return GetTypeName(edmType, isNullable: null, modelNamespace: modelNamespace);

}

public string GetTypeName(EdmType edmType, bool? isNullable, string modelNamespace)

{

if (edmType == null)

{

return null;

}

var collectionType = edmType as CollectionType;

if (collectionType != null)

{

return String.Format(CultureInfo.InvariantCulture, "ICollection<{0}>", GetTypeName(collectionType.TypeUsage, modelNamespace));

}

var typeName = \_code.Escape(edmType.MetadataProperties

.Where(p => p.Name == ExternalTypeNameAttributeName)

.Select(p => (string)p.Value)

.FirstOrDefault())

?? (modelNamespace != null && edmType.NamespaceName != modelNamespace ?

\_code.CreateFullName(\_code.EscapeNamespace(edmType.NamespaceName), \_code.Escape(edmType)) :

\_code.Escape(edmType));

if (edmType is StructuralType)

{

return typeName;

}

if (edmType is SimpleType)

{

var clrType = UnderlyingClrType(edmType);

if (!IsEnumType(edmType))

{

typeName = \_code.Escape(clrType);

}

typeName = FixNamespaces(typeName);

return clrType.IsValueType && isNullable == true ?

String.Format(CultureInfo.InvariantCulture, "Nullable<{0}>", typeName) :

typeName;

}

throw new ArgumentException("edmType");

}

public Type UnderlyingClrType(EdmType edmType)

{

ArgumentNotNull(edmType, "edmType");

var primitiveType = edmType as PrimitiveType;

if (primitiveType != null)

{

return primitiveType.ClrEquivalentType;

}

if (IsEnumType(edmType))

{

return GetEnumUnderlyingType(edmType).ClrEquivalentType;

}

return typeof(object);

}

public object GetEnumMemberValue(MetadataItem enumMember)

{

ArgumentNotNull(enumMember, "enumMember");

var valueProperty = enumMember.GetType().GetProperty("Value");

return valueProperty == null ? null : valueProperty.GetValue(enumMember, null);

}

public string GetEnumMemberName(MetadataItem enumMember)

{

ArgumentNotNull(enumMember, "enumMember");

var nameProperty = enumMember.GetType().GetProperty("Name");

return nameProperty == null ? null : (string)nameProperty.GetValue(enumMember, null);

}

public System.Collections.IEnumerable GetEnumMembers(EdmType enumType)

{

ArgumentNotNull(enumType, "enumType");

var membersProperty = enumType.GetType().GetProperty("Members");

return membersProperty != null

? (System.Collections.IEnumerable)membersProperty.GetValue(enumType, null)

: Enumerable.Empty<MetadataItem>();

}

public bool EnumIsFlags(EdmType enumType)

{

ArgumentNotNull(enumType, "enumType");

var isFlagsProperty = enumType.GetType().GetProperty("IsFlags");

return isFlagsProperty != null && (bool)isFlagsProperty.GetValue(enumType, null);

}

public bool IsEnumType(GlobalItem edmType)

{

ArgumentNotNull(edmType, "edmType");

return edmType.GetType().Name == "EnumType";

}

public PrimitiveType GetEnumUnderlyingType(EdmType enumType)

{

ArgumentNotNull(enumType, "enumType");

return (PrimitiveType)enumType.GetType().GetProperty("UnderlyingType").GetValue(enumType, null);

}

public string CreateLiteral(object value)

{

if (value == null || value.GetType() != typeof(TimeSpan))

{

return \_code.CreateLiteral(value);

}

return string.Format(CultureInfo.InvariantCulture, "new TimeSpan({0})", ((TimeSpan)value).Ticks);

}

public bool VerifyCaseInsensitiveTypeUniqueness(IEnumerable<string> types, string sourceFile)

{

ArgumentNotNull(types, "types");

ArgumentNotNull(sourceFile, "sourceFile");

var hash = new HashSet<string>(StringComparer.InvariantCultureIgnoreCase);

if (types.Any(item => !hash.Add(item)))

{

\_errors.Add(

new CompilerError(sourceFile, -1, -1, "6023",

String.Format(CultureInfo.CurrentCulture, CodeGenerationTools.GetResourceString("Template\_CaseInsensitiveTypeConflict"))));

return false;

}

return true;

}

public IEnumerable<SimpleType> GetEnumItemsToGenerate(IEnumerable<GlobalItem> itemCollection)

{

return GetItemsToGenerate<SimpleType>(itemCollection)

.Where(e => IsEnumType(e));

}

public IEnumerable<T> GetItemsToGenerate<T>(IEnumerable<GlobalItem> itemCollection) where T: EdmType

{

return itemCollection

.OfType<T>()

.Where(i => !i.MetadataProperties.Any(p => p.Name == ExternalTypeNameAttributeName))

.OrderBy(i => i.Name);

}

public IEnumerable<string> GetAllGlobalItems(IEnumerable<GlobalItem> itemCollection)

{

return itemCollection

.Where(i => i is EntityType || i is ComplexType || i is EntityContainer || IsEnumType(i))

.Select(g => GetGlobalItemName(g));

}

public string GetGlobalItemName(GlobalItem item)

{

if (item is EdmType)

{

return ((EdmType)item).Name;

}

else

{

return ((EntityContainer)item).Name;

}

}

public IEnumerable<EdmProperty> GetSimpleProperties(EntityType type)

{

return type.Properties.Where(p => p.TypeUsage.EdmType is SimpleType && p.DeclaringType == type);

}

public IEnumerable<EdmProperty> GetSimpleProperties(ComplexType type)

{

return type.Properties.Where(p => p.TypeUsage.EdmType is SimpleType && p.DeclaringType == type);

}

public IEnumerable<EdmProperty> GetComplexProperties(EntityType type)

{

return type.Properties.Where(p => p.TypeUsage.EdmType is ComplexType && p.DeclaringType == type);

}

public IEnumerable<EdmProperty> GetComplexProperties(ComplexType type)

{

return type.Properties.Where(p => p.TypeUsage.EdmType is ComplexType && p.DeclaringType == type);

}

public IEnumerable<EdmProperty> GetPropertiesWithDefaultValues(EntityType type)

{

return type.Properties.Where(p => p.TypeUsage.EdmType is SimpleType && p.DeclaringType == type && p.DefaultValue != null);

}

public IEnumerable<EdmProperty> GetPropertiesWithDefaultValues(ComplexType type)

{

return type.Properties.Where(p => p.TypeUsage.EdmType is SimpleType && p.DeclaringType == type && p.DefaultValue != null);

}

public IEnumerable<NavigationProperty> GetNavigationProperties(EntityType type)

{

return type.NavigationProperties.Where(np => np.DeclaringType == type);

}

public IEnumerable<NavigationProperty> GetCollectionNavigationProperties(EntityType type)

{

return type.NavigationProperties.Where(np => np.DeclaringType == type && np.ToEndMember.RelationshipMultiplicity == RelationshipMultiplicity.Many);

}

public FunctionParameter GetReturnParameter(EdmFunction edmFunction)

{

ArgumentNotNull(edmFunction, "edmFunction");

var returnParamsProperty = edmFunction.GetType().GetProperty("ReturnParameters");

return returnParamsProperty == null

? edmFunction.ReturnParameter

: ((IEnumerable<FunctionParameter>)returnParamsProperty.GetValue(edmFunction, null)).FirstOrDefault();

}

public bool IsComposable(EdmFunction edmFunction)

{

ArgumentNotNull(edmFunction, "edmFunction");

var isComposableProperty = edmFunction.GetType().GetProperty("IsComposableAttribute");

return isComposableProperty != null && (bool)isComposableProperty.GetValue(edmFunction, null);

}

public IEnumerable<FunctionImportParameter> GetParameters(EdmFunction edmFunction)

{

return FunctionImportParameter.Create(edmFunction.Parameters, \_code, \_ef);

}

public TypeUsage GetReturnType(EdmFunction edmFunction)

{

var returnParam = GetReturnParameter(edmFunction);

return returnParam == null ? null : \_ef.GetElementType(returnParam.TypeUsage);

}

public bool GenerateMergeOptionFunction(EdmFunction edmFunction, bool includeMergeOption)

{

var returnType = GetReturnType(edmFunction);

return !includeMergeOption && returnType != null && returnType.EdmType.BuiltInTypeKind == BuiltInTypeKind.EntityType;

}

}

public static void ArgumentNotNull<T>(T arg, string name) where T : class

{

if (arg == null)

{

throw new ArgumentNullException(name);

}

}

#>

//------------------------------------------------------------------------------

// <auto-generated>

// This code was generated from a template.

//

// Manual changes to this file may cause unexpected behavior in your application.

// Manual changes to this file will be overwritten if the code is regenerated.

// </auto-generated>

//------------------------------------------------------------------------------

namespace UniinfoAsp.Models

{

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

public partial class Chamado

{

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]

public Chamado()

{

this.chamadoAtendimentoes = new List<chamadoAtendimento>();

}

[Display(Name = "Chamado Id")]

public int idChamado { get; set; }

[Display(Name = "Funcionario Id")]

public int idFuncionario { get; set; }

[Display(Name = "Problema Id")]

public int idProblema { get; set; }

[Display(Name = "Descrição")]

public string descricao { get; set; }

[DataType(DataType.Date)]

[Display(Name = "Data de Abertura")]

public System.DateTime dataChamado { get; set; }

[Display(Name = "Status")]

public string statusAtendimento { get; set; }

public virtual Funcionario Funcionario { get; set; }

public virtual Problema Problema { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]

public virtual List<chamadoAtendimento> chamadoAtendimentoes { get; set; }

}

}

//------------------------------------------------------------------------------

// <auto-generated>

// This code was generated from a template.

//

// Manual changes to this file may cause unexpected behavior in your application.

// Manual changes to this file will be overwritten if the code is regenerated.

// </auto-generated>

//------------------------------------------------------------------------------

namespace UniinfoAsp.Models

{

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

public partial class chamadoAtendimento

{

public int idAtendimento { get; set; }

[Display(Name = "Funcionario Id")]

public int idFuncionario { get; set; }

[Display(Name = "Chamado Id")]

public int idChamado { get; set; }

[Display(Name = "Data de encerramento")]

public System.DateTime dataAtendimento { get; set; }

public virtual Chamado Chamado { get; set; }

public virtual Funcionario Funcionario { get; set; }

}

}

//------------------------------------------------------------------------------

// <auto-generated>

// This code was generated from a template.

//

// Manual changes to this file may cause unexpected behavior in your application.

// Manual changes to this file will be overwritten if the code is regenerated.

// </auto-generated>

//------------------------------------------------------------------------------

namespace UniinfoAsp.Models

{

using System;

using System.Collections.Generic;

public partial class Con\_Chamado

{

public string descricao { get; set; }

public System.DateTime dataChamado { get; set; }

public string statusAtendimento { get; set; }

public string nome { get; set; }

}

}

//------------------------------------------------------------------------------

// <auto-generated>

// This code was generated from a template.

//

// Manual changes to this file may cause unexpected behavior in your application.

// Manual changes to this file will be overwritten if the code is regenerated.

// </auto-generated>

//------------------------------------------------------------------------------

namespace UniinfoAsp.Models

{

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

public partial class Funcionario

{

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]

public Funcionario()

{

this.Chamadoes = new List<Chamado>();

this.chamadoAtendimentoes = new List<chamadoAtendimento>();

this.Loginns = new List<Loginn>();

}

[Display(Name = "Funcionario Id")]

public int idFuncionario { get; set; }

[Display(Name = "Nome")]

public string nome { get; set; }

[Display(Name = "Ramal")]

public Nullable<int> ramal { get; set; }

[Display(Name = "N° do computador")]

public Nullable<int> nComputador { get; set; }

[Display(Name = "Email")]

public string email { get; set; }

[Display(Name = "Setor")]

public string setor { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]

public virtual List<Chamado> Chamadoes { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]

public virtual List<chamadoAtendimento> chamadoAtendimentoes { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]

public virtual List<Loginn> Loginns { get; set; }

}

}

//------------------------------------------------------------------------------

// <auto-generated>

// This code was generated from a template.

//

// Manual changes to this file may cause unexpected behavior in your application.

// Manual changes to this file will be overwritten if the code is regenerated.

// </auto-generated>

//------------------------------------------------------------------------------

namespace UniinfoAsp.Models

{

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

public partial class Loginn

{

[Display(Name = "Login Id")]

public int idLogin { get; set; }

[Display(Name = "Funcionario Id")]

public int idFuncionario { get; set; }

[Display(Name = "Login")]

public string login { get; set; }

[Display(Name = "Senha")]

public string senha { get; set; }

[Display(Name = "Nivel de Acesso")]

public int idNivelAcesso { get; set; }

public virtual Funcionario Funcionario { get; set; }

public virtual nivelAcesso nivelAcesso { get; set; }

}

}

//------------------------------------------------------------------------------

// <auto-generated>

// This code was generated from a template.

//

// Manual changes to this file may cause unexpected behavior in your application.

// Manual changes to this file will be overwritten if the code is regenerated.

// </auto-generated>

//------------------------------------------------------------------------------

namespace UniinfoAsp.Models

{

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

public partial class nivelAcesso

{

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]

public nivelAcesso()

{

this.Loginns = new List<Loginn>();

}

[Display(Name = "Nivel Acesso Id")]

public int idNivelAcesso { get; set; }

[Display(Name = "Tipo de acesso")]

public string tipoAcesso { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]

public virtual List<Loginn> Loginns { get; set; }

}

}

//------------------------------------------------------------------------------

// <auto-generated>

// This code was generated from a template.

//

// Manual changes to this file may cause unexpected behavior in your application.

// Manual changes to this file will be overwritten if the code is regenerated.

// </auto-generated>

//------------------------------------------------------------------------------

namespace UniinfoAsp.Models

{

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

public partial class Problema

{

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]

public Problema()

{

this.Chamadoes = new List<Chamado>();

}

[Display(Name = "Problema Id")]

public int idProblema { get; set; }

[Display(Name = "Tipo de problema")]

public string tipoProblema { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]

public virtual List<Chamado> Chamadoes { get; set; }

}

}

//------------------------------------------------------------------------------

// <auto-generated>

// This code was generated from a template.

//

// Manual changes to this file may cause unexpected behavior in your application.

// Manual changes to this file will be overwritten if the code is regenerated.

// </auto-generated>

//------------------------------------------------------------------------------

**VIEWS**

**Account**

@model UniinfoAsp.Models.Loginn

@{

/\*\*/

ViewBag.Title = "Entrar";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<section class="inner-banner" style="background-color:#fff;">

<div class="container">

</div>

</section>

<section class="signin py-5" style="background-color:#fff; margin-top:-10px;">

<div class="container">

<div class="row main-content-agile">

<div class="col-lg-6 col-md-9 mx-auto">

<div class="sub-main-w3 text-center">

<h3>Entrar</h3>

<hr />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="icon1">

@Html.LabelFor(model => model.login, "", htmlAttributes: new { @class = "control-label col-md-2" })

<div>

@Html.EditorFor(model => model.login, new { htmlAttributes = new { @placeholder = "Login", style = "text-align:center;color: #0042fe;" } })

@Html.ValidationMessageFor(model => model.login, "", new { @class = "text-danger" })

</div>

</div>

<div class="icon2">

@Html.LabelFor(model => model.senha, "", htmlAttributes: new { @class = "control-label col-md-2" })

<div>

@Html.EditorFor(model => model.senha, new { htmlAttributes = new { @placeholder = "Senha", style = "text-align:center;color: #0042fe;" } })

@Html.ValidationMessageFor(model => model.senha, "", new { @class = "text-danger" })

</div>

</div>

<input type="submit" value="Login" class="btn btn-default" />

<div style=" padding-top:10px; ">

@ViewBag.Message

</div>

</div>

</div>

<div class="col-lg-6 col-md-8 mx-auto">

<img src="~/Content/images/banner.png" alt="" class="img-fluid" />

</div>

</div>

</div>

</section>

}

**Admin**

@{

ViewBag.Title = "Inicio";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

<body>

<!-- banner -->

<section class="banner" style="background-color:#fff;">

<div class="container">

<div class="row banner-grids">

<div class="col-lg-6 banner-info">

<h3 class="mb-3">Uniinfo</h3>

<p class="mb-4">

Bem vindo Administrador. <br/>

Vamos começar!

</p>

</div>

<div class="col-lg-6 col-md-9 banner-image">

<img src="~/Content/images/pcroxo.jpg" alt="" class="img-fluid" />

</div>

</div>

</div>

</section>

<!-- //banner -->

<!-- banner2 -->

<section class="banner" style="background-color:#f5f5f5;">

<div class="container">

<div class="row banner-grids">

<div class="col-lg-6 banner-info">

<h3 class="mb-3" style="color:#0042fe;">Atenção</h3>

<p class="mb-4">

Esteja atento ao editar qualquer informação.<br/>

informações erradas podem prejudicar <br/>

o funcionamento de todo sistema

</p>

</div>

<div class="col-lg-6 col-md-9 banner-image">

<img src="~/Content/images/café.png" alt="" class="img-fluid" />

</div>

</div>

</div>

</section>

<!-- //banner2 -->

</body>

**Chamado**

@{

ViewBag.Title = "Inicio";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

<body>

<!-- banner -->

<section class="banner" style="background-color:#fff;">

<div class="container">

<div class="row banner-grids">

<div class="col-lg-6 banner-info">

<h3 class="mb-3">Uniinfo</h3>

<p class="mb-4">

Bem vindo Administrador. <br/>

Vamos começar!

</p>

</div>

<div class="col-lg-6 col-md-9 banner-image">

<img src="~/Content/images/pcroxo.jpg" alt="" class="img-fluid" />

</div>

</div>

</div>

</section>

<!-- //banner -->

<!-- banner2 -->

<section class="banner" style="background-color:#f5f5f5;">

<div class="container">

<div class="row banner-grids">

<div class="col-lg-6 banner-info">

<h3 class="mb-3" style="color:#0042fe;">Atenção</h3>

<p class="mb-4">

Esteja atento ao editar qualquer informação.<br/>

informações erradas podem prejudicar <br/>

o funcionamento de todo sistema

</p>

</div>

<div class="col-lg-6 col-md-9 banner-image">

<img src="~/Content/images/café.png" alt="" class="img-fluid" />

</div>

</div>

</div>

</section>

<!-- //banner2 -->

</body>

@model UniinfoAsp.Models.Chamado

@{

ViewBag.Title = "Posicionar";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

@Html.ViewBag

<div style="margin:90px 250px 20px 320px;">

<h4>Posicionar</h4>

<hr />

<dl>

<dt>

@Html.DisplayNameFor(model => model.descricao)

</dt>

<dd>

@Html.DisplayFor(model => model.descricao)

</dd>

<dt>

@Html.DisplayNameFor(model => model.dataChamado)

</dt>

<dd>

@Html.DisplayFor(model => model.dataChamado)

</dd>

<dt>

@Html.DisplayNameFor(model => model.statusAtendimento)

</dt>

<dd>

@Html.DisplayFor(model => model.statusAtendimento)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Funcionario.nome)

</dt>

<dd>

@Html.DisplayFor(model => model.Funcionario.nome)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Problema.tipoProblema)

</dt>

<dd>

@Html.DisplayFor(model => model.Problema.tipoProblema)

</dd>

</dl>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-actions no-color">

<input type="submit" value="Posicionar" class="btn btn-default" />

</div>

}

<div class="btn btn-primary" style="margin-top:5px;">

@Html.ActionLink("Voltar", "Index", null, new { style = "color:white;" })

</div>

</div>

@model UniinfoAsp.Models.Chamado

@{

ViewBag.Title = "Detalhes";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

@Html.ViewBag

<div style="margin:90px 250px 20px 320px;">

<h4>Detalhes</h4>

<hr />

<dl>

<dt>

@Html.DisplayNameFor(model => model.Funcionario.nome)

</dt>

<dd>

@Html.DisplayFor(model => model.Funcionario.nome)

</dd>

<dt>

@Html.DisplayNameFor(model => model.descricao)

</dt>

<dd>

@Html.DisplayFor(model => model.descricao)

</dd>

<dt>

@Html.DisplayNameFor(model => model.dataChamado)

</dt>

<dd>

@Html.DisplayFor(model => model.dataChamado)

</dd>

<dt>

@Html.DisplayNameFor(model => model.statusAtendimento)

</dt>

<dd>

@Html.DisplayFor(model => model.statusAtendimento)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Problema.tipoProblema)

</dt>

<dd>

@Html.DisplayFor(model => model.Problema.tipoProblema)

</dd>

</dl>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-actions no-color">

<input type="submit" value="Deletar" class="btn btn-default" />

</div>

}

<div class="btn btn-primary" style="margin-top:5px;">

@Html.ActionLink("Voltar", "Index", null, new { style = "color:white;" })

</div>

</div>

@model UniinfoAsp.Models.Chamado

@{

ViewBag.Title = "Deletar";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

@Html.ViewBag

<div style="margin:90px 250px 20px 320px;">

<h4>Deletar</h4>

<hr />

<dl>

<dt>

@Html.DisplayNameFor(model => model.descricao)

</dt>

<dd>

@Html.DisplayFor(model => model.descricao)

</dd>

<dt>

@Html.DisplayNameFor(model => model.dataChamado)

</dt>

<dd>

@Html.DisplayFor(model => model.dataChamado)

</dd>

<dt>

@Html.DisplayNameFor(model => model.statusAtendimento)

</dt>

<dd>

@Html.DisplayFor(model => model.statusAtendimento)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Funcionario.nome)

</dt>

<dd>

@Html.DisplayFor(model => model.Funcionario.nome)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Problema.tipoProblema)

</dt>

<dd>

@Html.DisplayFor(model => model.Problema.tipoProblema)

</dd>

</dl>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-actions no-color">

<input type="submit" value="Deletar" class="btn btn-default" />

</div>

}

<div class="btn btn-primary" style="margin-top:5px;">

@Html.ActionLink("Voltar", "Index", null, new { style = "color:white;" })

</div>

</div>

**ChamadoAtendimento**

@model IEnumerable<UniinfoAsp.Models.chamadoAtendimento>

@{

ViewBag.Title = "Chamados";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

@Html.ViewBag

<div style="margin-left:100px; margin-right:100px;">

<h2 style="margin:90px 0px 55px 0px;">Chamados em atendimento</h2>

<table class="table">

<tr>

<th>

@Html.DisplayNameFor(model => model.Funcionario.nome)

</th>

<th>

@Html.DisplayNameFor(model => model.Funcionario.email)

</th>

<th>

@Html.DisplayNameFor(model => model.Chamado.statusAtendimento)

</th>

<th>

@Html.DisplayNameFor(model => model.Chamado.Problema.tipoProblema)

</th>

<th></th>

</tr>

@foreach (var item in Model)

{

<tr>

<td>

@Html.DisplayFor(modelItem => item.Funcionario.nome)

</td>

<td>

@Html.DisplayFor(model => item.Funcionario.email)

</td>

<td>

@Html.DisplayFor(modelItem => item.Chamado.statusAtendimento)

</td>

<td>

@Html.DisplayFor(modelItem => item.Chamado.Problema.tipoProblema)

</td>

<td class="btn btn-primary" style=" margin:0px 5px 5px 10px;">

@Html.ActionLink("Detalhes", "Details", new { id = item.idAtendimento }, new { style = "color:white; padding:10px;" })

</td>

<td class="btn btn-primary">

@Html.ActionLink("Deletar", "Delete", new { id = item.idAtendimento }, new { style = "color:white; padding:10px;" })

</td>

<td class="btn btn-primary" style="margin-left:5px;">

@Html.ActionLink("Encerrar", "Encerrar", new { id = item.idAtendimento }, new { style = "color:white; padding:10px;" })

</td>

</tr>

}

</table>

</div>

@model UniinfoAsp.Models.chamadoAtendimento

@{

ViewBag.Title = "Encerrar";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

@Html.ViewBag

<div style="margin:90px 250px 20px 320px;">

<h4>Deletar</h4>

<hr />

<dl>

<dt>

@Html.DisplayNameFor(model => model.Chamado.dataChamado)

</dt>

<dd>

@Html.DisplayFor(model => model.Chamado.dataChamado)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Chamado.descricao)

</dt>

<dd>

@Html.DisplayFor(model => model.Chamado.descricao)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Funcionario.nome)

</dt>

<dd>

@Html.DisplayFor(model => model.Funcionario.nome)

</dd>

</dl>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-actions no-color">

<input type="submit" value="Encerrar" class="btn btn-default" />

</div>

}

<div class="btn btn-primary" style="margin-top:5px;">

@Html.ActionLink("Voltar", "Index", null, new { style = "color:white;" })

</div>

</div>

@model UniinfoAsp.Models.chamadoAtendimento

@{

ViewBag.Title = "Detalhes";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

@Html.ViewBag

<div style="margin:90px 250px 20px 320px;">

<h4>Deletar</h4>

<hr />

<dl>

<dt>

@Html.DisplayNameFor(model => model.Chamado.dataChamado)

</dt>

<dd>

@Html.DisplayFor(model => model.Chamado.dataChamado)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Chamado.Problema.tipoProblema)

</dt>

<dd>

@Html.DisplayFor(model => model.Chamado.Problema.tipoProblema)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Chamado.descricao)

</dt>

<dd>

@Html.DisplayFor(model => model.Chamado.descricao)

</dd>

</dl>

<div class="form-actions no-color">

<input type="submit" value="Deletar" class="btn btn-default" />

</div>

<div class="btn btn-primary" style="margin-top:5px;">

@Html.ActionLink("Voltar", "Index", null, new { style = "color:white;" })

</div>

</div>

@model UniinfoAsp.Models.chamadoAtendimento

@{

ViewBag.Title = "Deletar";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

@Html.ViewBag

<div style="margin:90px 250px 20px 320px;">

<h4>Deletar</h4>

<hr />

<dl>

<dt>

@Html.DisplayNameFor(model => model.Chamado.idChamado)

</dt>

<dd>

@Html.DisplayFor(model => model.Chamado.idChamado)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Funcionario.nome)

</dt>

<dd>

@Html.DisplayFor(model => model.Funcionario.nome)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Chamado.Problema.tipoProblema)

</dt>

<dd>

@Html.DisplayFor(model => model.Chamado.Problema.tipoProblema)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Chamado.descricao)

</dt>

<dd>

@Html.DisplayFor(model => model.Chamado.descricao)

</dd>

</dl>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-actions no-color">

<input type="submit" value="Deletar" class="btn btn-default" />

</div>

}

<div class="btn btn-primary" style="margin-top:5px;">

@Html.ActionLink("Voltar", "Index", null, new { style = "color:white;" })

</div>

</div>

**Contato**

@{

ViewBag.Title = "Index";

Layout = "~/Views/Shared/\_LayoutFuncionario.cshtml";

}

@Html.ViewBag

<div style="margin-top:-20px;">

<!-- banner -->

<section class="banner" style="background-color:#fff;">

<div class="container">

<div class="row banner-grids">

<div class="col-lg-6 banner-info">

<h3 class="mb-3">Problemas?</h3>

<p class="mb-4">

Caso não tenha sido atendido ainda contate-nos <br>

e aguarde que o responderemos o<br>

mais breve possível.

</p>

</div>

<div class="col-lg-6 col-md-9 banner-image">

<img src="~/Content/images/webpc.png" alt="" class="img-fluid" />

</div>

</div>

</div>

</section>

<!-- //banner -->

<!--Contate-nos-->

<section class="security bg-light py-5">

<div class="container py-md-3">

<div class="security-bottom text-center">

<h3 class="my-4" style="color:#0042fe;">Entre em contato!</h3>

<p class="">Você pode nos contatar através do nosso email ou pelo telefone da empresa</p>

<p class="my-4">

Administrador@gmail.com<br>

(015) 3593-0456

</p>

</div>

</div>

</section>

<!--Contate-nos-->

</div>

**Func**

@{

ViewBag.Title = "Index";

Layout = "~/Views/Shared/\_LayoutFuncionario.cshtml";

}

<body>

<!-- banner -->

<section class="banner" style="background-color:#fff;">

<div class="container">

<div class="row banner-grids">

<div class="col-lg-6 banner-info">

<h3 class="mb-3">Uniinfo</h3>

<p class="mb-4">

Bem vindo! <br />

Cadastre seu problema em nosso sistema

</p>

</div>

<div class="col-lg-6 col-md-9 banner-image">

<div style="height:50px;"></div>

<img src="~/Content/images/helpdesk\_logo\_nbg\_default\_menor.png" alt="" class="img-fluid" />

</div>

</div>

</div>

</section>

<!-- //banner -->

<!-- banner2 -->

<section class="banner" style="background-color:#f5f5f5;">

<div class="container">

<div class="row banner-grids">

<div class="col-lg-6 banner-info" style="margin-top:-50px;">

<h3 class="mb-3" style="color:#0042fe;">Atenção</h3>

<p class="mb-4">

Esteja atento ao cadastrar qualquer informação.<br />

informações erradas podem prejudicar <br />

o atendimento do seu problema.

</p>

</div>

<div class="col-lg-6 col-md-9 banner-image">

<div style="height:30px;"></div>

<img src="~/Content/images/helpdesk\_logo\_bg\_blue\_m2.png" alt="" class="img-fluid" />

</div>

</div>

</div>

</section>

<!-- //banner2 -->

</body>

@model UniinfoAsp.Models.Chamado

@{

/\*\*/

ViewBag.Title = "CadastrarChamado";

Layout = "~/Views/Shared/\_LayoutFuncionario.cshtml";

}

@Html.ViewBag

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-horizontal" style="margin:90px 250px 20px 320px;">

<h2 style="margin-bottom:15px;">Digite as informações</h2>

<h4>Chamado</h4>

<hr />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.idFuncionario, "idFuncionario", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.DropDownList("idFuncionario", new SelectList(ViewBag.funcionarioId, "idFuncionario", "idFuncionario"), "id", htmlAttributes: new { @class = "form-control" })

@Html.ValidationMessageFor(model => model.idFuncionario, "", new { @class = "text-danger" })

</div>

</div>

@\*<div class="form-group">

@Html.LabelFor(model => model.idFuncionario, "Nome Funcionario", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.DropDownList("idFuncionario", null, htmlAttributes: new { @class = "form-control" })

@Html.ValidationMessageFor(model => model.idFuncionario, "", new { @class = "text-danger" })

</div>

</div>\*@

<div class="form-group">

@Html.LabelFor(model => model.idProblema, "Problema", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.DropDownList("idProblema", new SelectList(ViewBag.problema, "idProblema", "tipoProblema"), "Selecione um Problema", htmlAttributes: new { @class = "form-control" })

@Html.ValidationMessageFor(model => model.idProblema, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.descricao, "Descrição", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.descricao, new { htmlAttributes = new { @maxlength = "90", @class = "form-control" } })

@Html.ValidationMessageFor(model => model.descricao, "", new { @class = "text-danger" })

</div>

</div>

@\*<div class="form-group">

@Html.LabelFor(model => model.dataChamado, "Data Chamado", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.TextBoxFor(model => model.dataChamado, new { @Value = String.Format("{0:dd/MM/yyyy}", System.DateTime.Today) })

</div>

</div>\*@

@\*<div class="form-group">

@Html.LabelFor(model => model.statusAtendimento, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.TextBoxFor(model => model.statusAtendimento, new { @Value = String.Format("Aberto") })

</div>

</div>\*@

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<input type="submit" value="Cadastrar" class="btn btn-default" />

</div>

</div>

</div>

}

@section Scripts {

@Scripts.Render("~/bundles/jqueryval")

}

**Funcionario**

@model IEnumerable<UniinfoAsp.Models.Funcionario>

@{

ViewBag.Title = "Funcionarios";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

@Html.ViewBag

<div style="margin:0px 100px;">

<h2 style="margin:90px 0px 10px 0px;">Funcionarios</h2>

<p>

@Html.ActionLink("Cadastrar novo funcionario", "Create")

</p>

<table class="table">

<tr>

<th>

@Html.DisplayNameFor(model => model.nome)

</th>

<th>

@Html.DisplayNameFor(model => model.ramal)

</th>

<th>

@Html.DisplayNameFor(model => model.nComputador)

</th>

<th>

@Html.DisplayNameFor(model => model.email)

</th>

<th>

@Html.DisplayNameFor(model => model.setor)

</th>

<th></th>

</tr>

@foreach (var item in Model)

{

<tr>

<td>

@Html.DisplayFor(modelItem => item.nome)

</td>

<td>

@Html.DisplayFor(modelItem => item.ramal)

</td>

<td>

@Html.DisplayFor(modelItem => item.nComputador)

</td>

<td>

@Html.DisplayFor(modelItem => item.email)

</td>

<td>

@Html.DisplayFor(modelItem => item.setor)

</td>

<td class="btn btn-primary" style=" margin:0px 5px 5px 10px;">

@Html.ActionLink("Editar", "Edit", new { id = item.idFuncionario }, new {style = "color:white; padding:10px;"})

</td>

<td class="btn btn-primary">

@Html.ActionLink("Deletar", "Delete", new { id = item.idFuncionario }, new { style = "color : white; padding:10px;" })

</td>

</tr>

}

</table>

</div>

@model UniinfoAsp.Models.Funcionario

@{

ViewBag.Title = "Editar";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

@Html.ViewBag

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div style="margin:90px 250px 20px 320px;">

<h4>Editar</h4>

<hr />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

@Html.HiddenFor(model => model.idFuncionario)

<div class="form-group">

@Html.LabelFor(model => model.nome, "Nome", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.nome, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.nome, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.ramal, "Ramal", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.ramal, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.ramal, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.nComputador, "Nº Computador", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.nComputador, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.nComputador, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.email, "E-mail", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.email, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.email, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.setor, "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="Salvar" class="btn btn-default" />

</div>

</div>

<div class="btn btn-primary" style="margin-left:15px;">

@Html.ActionLink("Voltar", "Index", null, new { style = "color:white;" })

</div>

</div>

}

@section Scripts {

@Scripts.Render("~/bundles/jqueryval")

}

@model UniinfoAsp.Models.Funcionario

@{

ViewBag.Title = "Deletar";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

@Html.ViewBag

<div style="margin:90px 250px 20px 320px;">

<h4>Deletar</h4>

<hr />

<dl>

<dt>

@Html.DisplayNameFor(model => model.nome)

</dt>

<dd>

@Html.DisplayFor(model => model.nome)

</dd>

<dt>

@Html.DisplayNameFor(model => model.ramal)

</dt>

<dd>

@Html.DisplayFor(model => model.ramal)

</dd>

<dt>

@Html.DisplayNameFor(model => model.nComputador)

</dt>

<dd>

@Html.DisplayFor(model => model.nComputador)

</dd>

<dt>

@Html.DisplayNameFor(model => model.email)

</dt>

<dd>

@Html.DisplayFor(model => model.email)

</dd>

<dt>

@Html.DisplayNameFor(model => model.setor)

</dt>

<dd>

@Html.DisplayFor(model => model.setor)

</dd>

</dl>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div>

<input type="submit" value="Deletar" class="btn btn-default" />

</div>

}

<div class="btn btn-primary" style="margin-top:5px;">

@Html.ActionLink("Voltar", "Index", null, new { style = "color:white;" })

</div>

<br>

<br>

<p class="mb-4">

@ViewBag.message

</p>

</div>

@model UniinfoAsp.Models.Funcionario

@{

ViewBag.Title = "Cadastrar Funcionário";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

@Html.ViewBag

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div style="margin:90px 250px 20px 320px;">

<h2 style="margin-bottom:15px">Digite as informações</h2>

<h4>Funcionario</h4>

<hr />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.nome, "Nome", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.nome, new { htmlAttributes = new {@maxlength="30", @class = "form-control" } })

@Html.ValidationMessageFor(model => model.nome, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.ramal, "Ramal", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.ramal, new { htmlAttributes = new { @maxlength = "30", @class = "form-control" }})

@Html.ValidationMessageFor(model => model.ramal, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.nComputador, "Nº Computador", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.nComputador, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.nComputador, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.email, "E-mail", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.email, new { htmlAttributes = new { @maxlength = "50", @class = "form-control" } })

@Html.ValidationMessageFor(model => model.email, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.setor, "Setor", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.setor, new { htmlAttributes = new { @maxlength = "30", @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" />

</div>

</div>

<div class="btn btn-primary" style="margin-left:15px;">

@Html.ActionLink("Voltar", "Index",null,new { style = "color:white;"})

</div>

</div>

}

@section Scripts {

@Scripts.Render("~/bundles/jqueryval")

}

**Home**

@{

ViewBag.Title = "Início";

}

<body>

<!-- banner -->

<section class="banner" style="background-color:#fff;">

<div class="container">

<div class="row banner-grids">

<div class="col-lg-6 banner-info">

<h3 class="mb-3">Uniinfo</h3>

<p class="mb-4">

Bem Vindo ao Uniinfo-HelpDesk, é um prazer te-lo conosco!.

</p>

@Html.ActionLink("Sobre nós", "Index", "Sobre")

</div>

<div class="col-lg-6 col-md-9 banner-image">

<img src="~/Content/images/banner.png" alt="" class="img-fluid" />

</div>

</div>

</div>

</section>

<!-- //banner -->

<section class="about-us bg-light py-5">

<div class="container py-md-3">

<div class="row bottom-grids">

<div class="col-lg-4 col-md-5 col-6 px-lg-5">

<img src="~/Content/images/webpc.png" alt="" class="img-fluid" />

</div>

<div class="col-lg-6 offset-lg-2 col-md-7 text-right px-lg-5 mt-md-0 mt-4">

<h3 class="my-3" style="color:#0042fe;">Torne seu problema nosso problema.</h3>

<p class="mb-4">

O Uniinfo é uma ferramenta prática e ágil que visa fornecer soluções de problemas,

tornando os conhecidos pelo setor responsável.

</p>

@Html.ActionLink("Login", "Login", "Account")

</div>

</div>

</div>

</section>

</body>

**Loginn**

@model IEnumerable<UniinfoAsp.Models.Loginn>

@{

ViewBag.Title = "Login";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

@Html.ViewBag

<div style="margin:0px 100px;">

<h2 style="margin:90px 0px 10px 0px;">Logins</h2>

<p>

@Html.ActionLink("Criar novo Login", "Create")

</p>

<table class="table">

<tr>

<th>

@Html.DisplayNameFor(model => model.login)

</th>

<th>

@Html.DisplayNameFor(model => model.Funcionario.nome)

</th>

<th>

@Html.DisplayNameFor(model => model.Funcionario.nComputador)

</th>

<th>

@Html.DisplayNameFor(model => model.nivelAcesso.tipoAcesso)

</th>

<th></th>

<th></th>

</tr>

@foreach (var item in Model)

{

<tr>

<td>

@Html.DisplayFor(modelItem => item.login)

</td>

<td>

@Html.DisplayFor(modelItem => item.Funcionario.nome)

</td>

<td>

@Html.DisplayFor(modelItem => item.Funcionario.nComputador)

</td>

<td>

@Html.DisplayFor(modelItem => item.nivelAcesso.tipoAcesso)

</td>

<td></td>

<td class="btn btn-primary" style=" margin:0px 5px 5px 230px;">

@Html.ActionLink("Editar", "Edit", new { id = item.idLogin }, new { style = "color:white; padding:10px;" })

</td>

<td class="btn btn-primary">

@Html.ActionLink("Detalhes", "Details", new { id = item.idLogin }, new { style = "color : white; padding:10px;" })

</td>

<td class="btn btn-primary" style="margin-left:5px;">

@Html.ActionLink("Deletar", "Delete", new { id = item.idLogin }, new { style = "color:white; padding:10px;" })

</td>

</tr>

}

</table>

</div>

@model UniinfoAsp.Models.Loginn

@{

ViewBag.Title = "Editar";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

@Html.ViewBag

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div style="margin:90px 250px 20px 320px;">

<h4>Login</h4>

<hr />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

@Html.HiddenFor(model => model.idLogin)

<div class="form-group">

@Html.LabelFor(model => model.Funcionario.idFuncionario, "Nome", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.DropDownList("idFuncionario", null, htmlAttributes: new { @class = "form-control" })

@Html.ValidationMessageFor(model => model.Funcionario.idFuncionario, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.login, "Login", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.login, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.login, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.senha, "Senha", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.senha, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.senha, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.idNivelAcesso, "Nivel Acesso", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.DropDownList("idNivelAcesso", null, htmlAttributes: new { @class = "form-control" })

@Html.ValidationMessageFor(model => model.idNivelAcesso, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<input type="submit" value="Salvar" class="btn btn-default" />

</div>

</div>

<div class="btn btn-primary" style="margin-left:15px;">

@Html.ActionLink("Voltar", "Index", null, new { style = "color:white;" })

</div>

</div>

}

@section Scripts {

@Scripts.Render("~/bundles/jqueryval")

}

@model UniinfoAsp.Models.Loginn

@{

ViewBag.Title = "Detalhes";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

@Html.ViewBag

<div style="margin:90px 250px 20px 320px;">

<h4>Detalhes</h4>

<hr />

<dl>

<dt>

@Html.DisplayNameFor(model => model.login)

</dt>

<dd>

@Html.DisplayFor(model => model.login)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Funcionario.email)

</dt>

<dd>

@Html.DisplayFor(model => model.Funcionario.email)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Funcionario.nComputador)

</dt>

<dd>

@Html.DisplayFor(model => model.Funcionario.nComputador)

</dd>

</dl>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-actions no-color">

<input type="submit" value="Deletar" class="btn btn-default" />

</div>

}

<div class="btn btn-primary" style="margin-top:5px;">

@Html.ActionLink("Voltar", "Index", null, new { style = "color:white;" })

</div>

<br>

<br>

<p class="mb-4">

@ViewBag.message

</p>

</div>

@model UniinfoAsp.Models.Loginn

@{

ViewBag.Title = "Deletar";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

@Html.ViewBag

<div style="margin:90px 250px 20px 320px;">

<h4>Deletar</h4>

<hr />

<dl>

<dt>

@Html.DisplayNameFor(model => model.login)

</dt>

<dd>

@Html.DisplayFor(model => model.login)

</dd>

<dt>

@Html.DisplayNameFor(model => model.senha)

</dt>

<dd>

@Html.DisplayFor(model => model.senha)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Funcionario.nome)

</dt>

<dd>

@Html.DisplayFor(model => model.Funcionario.nome)

</dd>

<dt>

@Html.DisplayNameFor(model => model.nivelAcesso.tipoAcesso)

</dt>

<dd>

@Html.DisplayFor(model => model.nivelAcesso.tipoAcesso)

</dd>

</dl>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-actions no-color">

<input type="submit" value="Deletar" class="btn btn-default" />

</div>

}

<div class="btn btn-primary" style="margin-top:5px;">

@Html.ActionLink("Voltar", "Index", null, new { style = "color:white;" })

</div>

<br>

<br>

<p class="mb-4">

@ViewBag.message

</p>

</div>

@model UniinfoAsp.Models.Loginn

@{

ViewBag.Title = "Criar";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

@Html.ViewBag

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-horizontal" style="margin:90px 250px 20px 320px;">

<h2 style="margin-bottom:15px;">Digite as informações</h2>

<h4>Login</h4>

<hr />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.idFuncionario, "Nome", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.DropDownList("idFuncionario", null, htmlAttributes: new { @class = "form-control" })

@Html.ValidationMessageFor(model => model.idFuncionario, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.login, "Login", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.login, new { htmlAttributes = new { @maxlength = "50", @class = "form-control" } })

@Html.ValidationMessageFor(model => model.login, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.senha, "Senha", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.senha, new { htmlAttributes = new { @maxlength = "50", @class = "form-control" } })

@Html.ValidationMessageFor(model => model.senha, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.idNivelAcesso, "Nivel de acesso", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.DropDownList("idNivelAcesso", null, htmlAttributes: new { @class = "form-control" })

@Html.ValidationMessageFor(model => model.idNivelAcesso, "", 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" />

</div>

</div>

<div class="btn btn-primary" style="margin-left:15px;">

@Html.ActionLink("Voltar", "Index", null, new { style = "color:white;" })

</div>

</div>

}

@section Scripts {

@Scripts.Render("~/bundles/jqueryval")

}

**Problema**

@model IEnumerable<UniinfoAsp.Models.Problema>

@{

ViewBag.Title = "Problema";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

@Html.ViewBag

<div style="margin:0px 100px;">

<h2 style="margin:90px 0px 10px 0px;">Problema</h2>

<p>

@Html.ActionLink("Adicionar novo tipo de problema", "Create")

</p>

<table class="table">

<tr>

<th>

@Html.DisplayNameFor(model => model.tipoProblema)

</th>

<th></th>

</tr>

@foreach (var item in Model)

{

<tr>

<td>

@Html.DisplayFor(modelItem => item.tipoProblema)

</td>

<td class="btn btn-primary" style=" margin:0px 5px 5px 820px;">

@Html.ActionLink("Editar", "Edit", new { id = item.idProblema }, new { style = "color:white; padding:10px;" })

</td>

<td class="btn btn-primary">

@Html.ActionLink("Deletar", "Delete", new { id = item.idProblema }, new { style = "color : white; padding:10px;" })

</td>

</tr>

}

</table>

</div>

@model UniinfoAsp.Models.Problema

@{

ViewBag.Title = "Editar";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

@Html.ViewBag

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div style="margin:90px 250px 20px 320px;">

<h4>Problema</h4>

<hr />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

@Html.HiddenFor(model => model.idProblema)

<div class="form-group">

@Html.LabelFor(model => model.tipoProblema, "Tipo do Problema", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.tipoProblema, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.tipoProblema, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<input type="submit" value="Salvar" class="btn btn-default" />

</div>

</div>

<div class="btn btn-primary" style="margin-left:15px;">

@Html.ActionLink("Voltar", "Index", null, new { style = "color:white;" })

</div>

</div>

}

@section Scripts {

@Scripts.Render("~/bundles/jqueryval")

}

@model UniinfoAsp.Models.Problema

@{

ViewBag.Title = "Deletar";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

@Html.ViewBag

<div style="margin:90px 250px 20px 320px;">

<h4>Deletar</h4>

<hr />

<dl>

<dt>

@Html.DisplayNameFor(model => model.tipoProblema)

</dt>

<dd>

@Html.DisplayFor(model => model.tipoProblema)

</dd>

</dl>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-actions no-color">

<input type="submit" value="Deletar" class="btn btn-default" />

</div>

}

<div class="btn btn-primary" style="margin-top:5px;">

@Html.ActionLink("Voltar", "Index", null, new { style = "color:white;" })

</div>

<br>

<br>

<p class="mb-4">

@ViewBag.message

</p>

</div>

@model UniinfoAsp.Models.Problema

@{

ViewBag.Title = "Tipos de Problemas";

Layout = "~/Views/Shared/\_LayoutAdmin.cshtml";

}

@Html.ViewBag

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div style="margin:90px 250px 20px 320px;">

<h2 style="margin-bottom:15px">Digite as informações</h2>

<h4>Problema</h4>

<hr />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.tipoProblema, "Tipo de Problema", htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.tipoProblema, new { htmlAttributes = new { @maxlength = "30", @class = "form-control" } })

@Html.ValidationMessageFor(model => model.tipoProblema, "", 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" />

</div>

</div>

<div class="btn btn-primary" style="margin-left:15px;">

@Html.ActionLink("Voltar", "Index", null, new { style = "color:white;" })

</div>

</div>

}

@section Scripts {

@Scripts.Render("~/bundles/jqueryval")

}

**Shared**

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>@ViewBag.Title - Uniinfo</title>

@Styles.Render("~/Content/css")

@Scripts.Render("~/bundles/modernizr")

</head>

<body>

<header class="bg-white py-1">

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

<div class="container">

<nav class="navbar navbar-expand-lg navbar-light">

<h1>

<a class="navbar-brand"><img src="~/Content/images/icone.png" alt="" class="img-fluid" />@Html.ActionLink("Início", "Index", "Home", new { area = "" }, new { @class = "navbar-brand" })</a>

</h1>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav ml-lg-4 mr-auto">

<li>@Html.ActionLink("Entrar", "Login", "Account", null, new { @class = "nav-link" })</li>

<li>@Html.ActionLink("Sobre nós", "Index", "Sobre", null, new { @class = "nav-link" })</li>

</ul>

</div>

</nav>

</div>

</div>

</header>

@RenderBody()

<!-- copyright -->

<section style="padding:30px 0px;">

<div class="container">

<div class="row">

<div class="col-lg-7 col-md-9">

<p class="" style="margin-top:10px; margin-bottom:-10px;">

&copy; @DateTime.Now.Year - Uniinfo. Todos os direitos reservados.

</p>

</div>

<div class="col-lg-5 col-md-3">

<ul class="social-iconsv2 agileinfo d-flex">

<li>

<a href="#">

<i class="fab fa-facebook-square"></i>

</a>

</li>

</ul>

</div>

</div>

</div>

</section>

<!-- copyright -->

</body>

</html>

@Scripts.Render("~/bundles/jquery")

@Scripts.Render("~/bundles/bootstrap")

@RenderSection("scripts", required: false)

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>@ViewBag.Title - Uniinfo</title>

@Styles.Render("~/Content/css")

@Scripts.Render("~/bundles/modernizr")

</head>

<body>

<header class="bg-white py-1">

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

<div class="container">

<nav class="navbar navbar-expand-lg navbar-light">

<h1>

<a class="navbar-brand"><img src="~/Content/images/icone.png" alt="" class="img-fluid" />@Html.ActionLink("Início", "Index", "Admin", new { area = "" }, new { @class = "navbar-brand" })</a>

</h1>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav ml-lg-4 mr-auto">

<li>@Html.ActionLink("Funcionarios", "Index", "Funcionario", null, new { @class = "nav-link" })</li>

<li>@Html.ActionLink("Chamados", "Index", "Chamado", null, new { @class = "nav-link" })</li>

<li>@Html.ActionLink("Chamados em andamento", "Index", "ChamadoAtendimento", null, new { @class = "nav-link" })</li>

<li>@Html.ActionLink("Problemas", "Index", "Problema", null, new { @class = "nav-link" })</li>

<li>@Html.ActionLink("Logins", "Index", "Loginn", null, new { @class = "nav-link" })</li>

<li>@Html.ActionLink("Sair", "LogOut", "Account", null, new { @class = "nav-link" })</li>

</ul>

</div>

</nav>

</div>

</div>

</header>

@RenderBody()

<!-- copyright -->

<section style="padding:30px 0px;">

<div class="container">

<div class="row">

<div class="col-lg-7 col-md-9">

<p class="" style="margin-top:10px; margin-bottom:-10px;">

&copy; @DateTime.Now.Year - Uniinfo. Todos os direitos reservados.

</p>

</div>

<div class="col-lg-5 col-md-3">

<ul class="social-iconsv2 agileinfo d-flex">

<li>

<a href="#">

<i class="fab fa-facebook-square"></i>

</a>

</li>

</ul>

</div>

</div>

</div>

</section>

<!-- copyright -->

</body>

</html>

@Scripts.Render("~/bundles/jquery")

@Scripts.Render("~/bundles/bootstrap")

@RenderSection("scripts", required: false)

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>@ViewBag.Title - Uniinfo</title>

@Styles.Render("~/Content/css")

@Scripts.Render("~/bundles/modernizr")

</head>

<body>

<header class="bg-white py-1">

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

<div class="container">

<nav class="navbar navbar-expand-lg navbar-light">

<h1>

<a class="navbar-brand"><img src="~/Content/images/icone.png" alt="" class="img-fluid" />@Html.ActionLink("Início", "Index", "Func", new { area = "" }, new { @class = "navbar-brand" })</a>

</h1>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav ml-lg-4 mr-auto">

<li>@Html.ActionLink("Chamado", "CadastrarChamado", "Func", null, new { @class = "nav-link" })</li>

<li>@Html.ActionLink("Contato", "Index", "Contato", null, new { @class = "nav-link" })</li>

<li>@Html.ActionLink("Sair", "LogOut", "Account", null, new { @class = "nav-link" })</li>

</ul>

</div>

</nav>

</div>

</div>

</header>

</body>

</html>

@RenderBody()

<!-- copyright -->

<section style="padding:30px 0px;">

<div class="container">

<div class="row">

<div class="col-lg-7 col-md-9">

<p class="" style="margin-top:10px; margin-bottom:-10px;">

&copy; @DateTime.Now.Year - Uniinfo. Todos os direitos reservados.

</p>

</div>

<div class="col-lg-5 col-md-3">

<ul class="social-iconsv2 agileinfo d-flex">

<li>

<a href="#">

<i class="fab fa-facebook-square"></i>

</a>

</li>

</ul>

</div>

</div>

</div>

</section>

<!-- copyright -->

@Scripts.Render("~/bundles/jquery")

@Scripts.Render("~/bundles/bootstrap")

@RenderSection("scripts", required: false)

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<meta name="viewport" content="width=device-width" />

<title>Erro</title>

</head>

<body>

<hgroup>

<h1>Erro.</h1>

<h2>Ocorreu um erro ao processar sua solicitação.</h2>

</hgroup>

</body>

</html>

**Sobre**

@{

ViewBag.Title = "Index";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<body style="margin-left:-10px;">

<!-- banner -->

<section class="banner" style="background-color:#fff;">

<div class="container">

<div class="row banner-grids">

<div class="col-lg-6 banner-info">

<h3 class="mb-3">Help-Desk</h3>

<p class="mb-4">

O unninfo é um sistema help-desk onde são cadastrados chamados de problemas, esses problemas são mais facilmente administrados a partir desse sistema.

</p>

</div>

<div class="col-lg-6 col-md-9 banner-image">

<div style="margin-top:150px;"></div>

<p class="mb-4">

Estamos melhorando dia após dia, esse projeto foi inicado em 2018 e continua em desenvolvimento.

</p>

</div>

</div>

</div>

</section>

<!-- //banner -->

<!--Contate-nos-->

<section class="security bg-light py-5">

<div class="container py-md-3">

<div class="security-bottom text-center">

<h3 class="my-4" style="color:#0042fe;">Tem um Login?</h3>

<p class="">Caso não tenha, entre em contato com o setor de TI ou um dos meios de comunicação abaixo.</p>

<p class="my-4">

Administrador@gmail.com<br>

(015) 3593-0456

</p>

</div>

</div>

</section>

<!--Contate-nos-->

<!-- banner -->

<section class="banner" style="background-color:#fff;">

<div class="container">

<div class="row banner-grids">

<div class="col-lg-6 banner-info">

<p class="mb-4" style="margin-top:-60px;">

Estamos aqui pra te auxiliar diante de diversos tipos de problema.<br><br>

Buscamos melhorar nosso projeto a cada atualizãção, sua opinião é importante para nós.<br><br>

Caso hajam mais dúvidas sobre nosso software contate-nos para obter mais informações.

</p>

</div>

<div class="col-lg-6 col-md-9 banner-image">

<img src="~/Content/images/imguniinfo2.jpg" alt="" class="img-fluid"/>

</div>

</div>

</div>

</section>

<!-- //banner -->

</body>

**VIEWSTART**

@{

Layout = "~/Views/Shared/\_Layout.cshtml";

}

**WEBCONFIG**

<?xml version="1.0"?>

<configuration>

<configSections>

<sectionGroup name="system.web.webPages.razor" type="System.Web.WebPages.Razor.Configuration.RazorWebSectionGroup, System.Web.WebPages.Razor, Version=3.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35">

<section name="host" type="System.Web.WebPages.Razor.Configuration.HostSection, System.Web.WebPages.Razor, Version=3.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" requirePermission="false" />

<section name="pages" type="System.Web.WebPages.Razor.Configuration.RazorPagesSection, System.Web.WebPages.Razor, Version=3.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" requirePermission="false" />

</sectionGroup>

</configSections>

<system.web.webPages.razor>

<host factoryType="System.Web.Mvc.MvcWebRazorHostFactory, System.Web.Mvc, Version=5.2.3.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" />

<pages pageBaseType="System.Web.Mvc.WebViewPage">

<namespaces>

<add namespace="System.Web.Mvc" />

<add namespace="System.Web.Mvc.Ajax" />

<add namespace="System.Web.Mvc.Html" />

<add namespace="System.Web.Optimization"/>

<add namespace="System.Web.Routing" />

<add namespace="UniinfoAsp" />

</namespaces>

</pages>

</system.web.webPages.razor>

<appSettings>

<add key="webpages:Enabled" value="false" />

</appSettings>

<system.webServer>

<handlers>

<remove name="BlockViewHandler"/>

<add name="BlockViewHandler" path="\*" verb="\*" preCondition="integratedMode" type="System.Web.HttpNotFoundHandler" />

</handlers>

</system.webServer>

<system.web>

<compilation>

<assemblies>

<add assembly="System.Web.Mvc, Version=5.2.3.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" />

</assemblies>

</compilation>

</system.web>

</configuration>

**WEBSERVICE**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Services;

using UniinfoAsp.DAL;

using UniinfoAsp.Models;

namespace UniinfoAsp.WebSevice

{

/// <summary>

/// Descrição resumida de WebService

/// </summary>

[WebService(Namespace = "http://tempuri.org/")]

[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]

[System.ComponentModel.ToolboxItem(false)]

// Para permitir que esse serviço da web seja chamado a partir do script, usando ASP.NET AJAX, remova os comentários da linha a seguir.

// [System.Web.Script.Services.ScriptService]

public class WebService : System.Web.Services.WebService

{

chamadoDAO dao = new chamadoDAO();

[WebMethod]

public string HelloWorld()

{

return "Olá, Mundo";

}

[WebMethod]

public List<Con\_Chamado> ConsultarChamado()

{

return dao.Consultar();

}

[WebMethod]

public List<Chamado> Consultar2()

{

return dao.Consultar2();

}

[WebMethod]

public string verificanivel(string loginwpf, string senhawpf)

{

return dao.verificaLogin(loginwpf, senhawpf);

}

}

}

**APPLICATIONINSIGHTS.CONFIG**

<?xml version="1.0" encoding="utf-8"?>

<ApplicationInsights xmlns="http://schemas.microsoft.com/ApplicationInsights/2013/Settings">

<TelemetryInitializers>

<Add Type="Microsoft.ApplicationInsights.DependencyCollector.HttpDependenciesParsingTelemetryInitializer, Microsoft.AI.DependencyCollector"/>

<Add Type="Microsoft.ApplicationInsights.WindowsServer.AzureRoleEnvironmentTelemetryInitializer, Microsoft.AI.WindowsServer"/>

<Add Type="Microsoft.ApplicationInsights.WindowsServer.AzureWebAppRoleEnvironmentTelemetryInitializer, Microsoft.AI.WindowsServer"/>

<Add Type="Microsoft.ApplicationInsights.WindowsServer.BuildInfoConfigComponentVersionTelemetryInitializer, Microsoft.AI.WindowsServer"/>

<Add Type="Microsoft.ApplicationInsights.Web.WebTestTelemetryInitializer, Microsoft.AI.Web"/>

<Add Type="Microsoft.ApplicationInsights.Web.SyntheticUserAgentTelemetryInitializer, Microsoft.AI.Web">

<!-- Extended list of bots:

search|spider|crawl|Bot|Monitor|BrowserMob|BingPreview|PagePeeker|WebThumb|URL2PNG|ZooShot|GomezA|Google SketchUp|Read Later|KTXN|KHTE|Keynote|Pingdom|AlwaysOn|zao|borg|oegp|silk|Xenu|zeal|NING|htdig|lycos|slurp|teoma|voila|yahoo|Sogou|CiBra|Nutch|Java|JNLP|Daumoa|Genieo|ichiro|larbin|pompos|Scrapy|snappy|speedy|vortex|favicon|indexer|Riddler|scooter|scraper|scrubby|WhatWeb|WinHTTP|voyager|archiver|Icarus6j|mogimogi|Netvibes|altavista|charlotte|findlinks|Retreiver|TLSProber|WordPress|wsr-agent|http client|Python-urllib|AppEngine-Google|semanticdiscovery|facebookexternalhit|web/snippet|Google-HTTP-Java-Client-->

<Filters>search|spider|crawl|Bot|Monitor|AlwaysOn</Filters>

</Add>

<Add Type="Microsoft.ApplicationInsights.Web.ClientIpHeaderTelemetryInitializer, Microsoft.AI.Web"/>

<Add Type="Microsoft.ApplicationInsights.Web.OperationNameTelemetryInitializer, Microsoft.AI.Web"/>

<Add Type="Microsoft.ApplicationInsights.Web.OperationCorrelationTelemetryInitializer, Microsoft.AI.Web"/>

<Add Type="Microsoft.ApplicationInsights.Web.UserTelemetryInitializer, Microsoft.AI.Web"/>

<Add Type="Microsoft.ApplicationInsights.Web.AuthenticatedUserIdTelemetryInitializer, Microsoft.AI.Web"/>

<Add Type="Microsoft.ApplicationInsights.Web.AccountIdTelemetryInitializer, Microsoft.AI.Web"/>

<Add Type="Microsoft.ApplicationInsights.Web.SessionTelemetryInitializer, Microsoft.AI.Web"/>

</TelemetryInitializers>

<TelemetryModules>

<Add Type="Microsoft.ApplicationInsights.DependencyCollector.DependencyTrackingTelemetryModule, Microsoft.AI.DependencyCollector"/>

<Add Type="Microsoft.ApplicationInsights.Extensibility.PerfCounterCollector.PerformanceCollectorModule, Microsoft.AI.PerfCounterCollector">

<!--

Use the following syntax here to collect additional performance counters:

<Counters>

<Add PerformanceCounter="\Process(??APP\_WIN32\_PROC??)\Handle Count" ReportAs="Process handle count" />

...

</Counters>

PerformanceCounter must be either \CategoryName(InstanceName)\CounterName or \CategoryName\CounterName

NOTE: performance counters configuration will be lost upon NuGet upgrade.

The following placeholders are supported as InstanceName:

??APP\_WIN32\_PROC?? - instance name of the application process for Win32 counters.

??APP\_W3SVC\_PROC?? - instance name of the application IIS worker process for IIS/ASP.NET counters.

??APP\_CLR\_PROC?? - instance name of the application CLR process for .NET counters.

-->

</Add>

<Add Type="Microsoft.ApplicationInsights.Extensibility.PerfCounterCollector.QuickPulse.QuickPulseTelemetryModule, Microsoft.AI.PerfCounterCollector"/>

<Add Type="Microsoft.ApplicationInsights.WindowsServer.DeveloperModeWithDebuggerAttachedTelemetryModule, Microsoft.AI.WindowsServer"/>

<Add Type="Microsoft.ApplicationInsights.WindowsServer.UnhandledExceptionTelemetryModule, Microsoft.AI.WindowsServer"/>

<Add Type="Microsoft.ApplicationInsights.WindowsServer.UnobservedExceptionTelemetryModule, Microsoft.AI.WindowsServer"/>

<Add Type="Microsoft.ApplicationInsights.Web.RequestTrackingTelemetryModule, Microsoft.AI.Web">

<Handlers>

<!--

Add entries here to filter out additional handlers:

NOTE: handler configuration will be lost upon NuGet upgrade.

-->

<Add>System.Web.Handlers.TransferRequestHandler</Add>

<Add>Microsoft.VisualStudio.Web.PageInspector.Runtime.Tracing.RequestDataHttpHandler</Add>

<Add>System.Web.StaticFileHandler</Add>

<Add>System.Web.Handlers.AssemblyResourceLoader</Add>

<Add>System.Web.Optimization.BundleHandler</Add>

<Add>System.Web.Script.Services.ScriptHandlerFactory</Add>

<Add>System.Web.Handlers.TraceHandler</Add>

<Add>System.Web.Services.Discovery.DiscoveryRequestHandler</Add>

<Add>System.Web.HttpDebugHandler</Add>

</Handlers>

</Add>

<Add Type="Microsoft.ApplicationInsights.Web.ExceptionTrackingTelemetryModule, Microsoft.AI.Web"/>

</TelemetryModules>

<TelemetryProcessors>

<Add Type="Microsoft.ApplicationInsights.Extensibility.PerfCounterCollector.QuickPulse.QuickPulseTelemetryProcessor, Microsoft.AI.PerfCounterCollector"/>

<Add Type="Microsoft.ApplicationInsights.WindowsServer.TelemetryChannel.AdaptiveSamplingTelemetryProcessor, Microsoft.AI.ServerTelemetryChannel">

<MaxTelemetryItemsPerSecond>5</MaxTelemetryItemsPerSecond>

</Add>

</TelemetryProcessors>

<TelemetryChannel Type="Microsoft.ApplicationInsights.WindowsServer.TelemetryChannel.ServerTelemetryChannel, Microsoft.AI.ServerTelemetryChannel"/>

<!--

Learn more about Application Insights configuration with ApplicationInsights.config here:

http://go.microsoft.com/fwlink/?LinkID=513840

Note: If not present, please add <InstrumentationKey>Your Key</InstrumentationKey> to the top of this file.

--></ApplicationInsights>

**GLOBAL.ASAX**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Web.Optimization;

using System.Web.Routing;

namespace UniinfoAsp

{

public class MvcApplication : System.Web.HttpApplication

{

protected void Application\_Start()

{

AreaRegistration.RegisterAllAreas();

FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);

RouteConfig.RegisterRoutes(RouteTable.Routes);

BundleConfig.RegisterBundles(BundleTable.Bundles);

}

protected void Application\_BeginRequest()

{

Response.Cache.SetCacheability(HttpCacheability.NoCache);

Response.Cache.SetExpires(DateTime.UtcNow.AddHours(-1));

Response.Cache.SetNoStore();

}

}

}

**PACKAGES.CONFIG**

<?xml version="1.0" encoding="utf-8"?>

<packages>

<package id="Antlr" version="3.4.1.9004" targetFramework="net471" />

<package id="bootstrap" version="3.0.0" targetFramework="net471" />

<package id="EntityFramework" version="6.2.0" targetFramework="net471" />

<package id="jQuery" version="1.10.2" targetFramework="net471" />

<package id="jQuery.Validation" version="1.11.1" targetFramework="net471" />

<package id="Microsoft.ApplicationInsights" version="2.2.0" targetFramework="net471" />

<package id="Microsoft.ApplicationInsights.Agent.Intercept" version="2.0.6" targetFramework="net471" />

<package id="Microsoft.ApplicationInsights.DependencyCollector" version="2.2.0" targetFramework="net471" />

<package id="Microsoft.ApplicationInsights.PerfCounterCollector" version="2.2.0" targetFramework="net471" />

<package id="Microsoft.ApplicationInsights.Web" version="2.2.0" targetFramework="net471" />

<package id="Microsoft.ApplicationInsights.WindowsServer" version="2.2.0" targetFramework="net471" />

<package id="Microsoft.ApplicationInsights.WindowsServer.TelemetryChannel" version="2.2.0" targetFramework="net471" />

<package id="Microsoft.AspNet.Mvc" version="5.2.3" targetFramework="net471" />

<package id="Microsoft.AspNet.Mvc.pt-br" version="5.2.3" targetFramework="net471" />

<package id="Microsoft.AspNet.Razor" version="3.2.3" targetFramework="net471" />

<package id="Microsoft.AspNet.Razor.pt-br" version="3.2.3" targetFramework="net471" />

<package id="Microsoft.AspNet.Web.Optimization" version="1.1.3" targetFramework="net471" />

<package id="Microsoft.AspNet.Web.Optimization.pt-br" version="1.1.3" targetFramework="net471" />

<package id="Microsoft.AspNet.WebPages" version="3.2.3" targetFramework="net471" />

<package id="Microsoft.AspNet.WebPages.pt-br" version="3.2.3" targetFramework="net471" />

<package id="Microsoft.CodeDom.Providers.DotNetCompilerPlatform" version="1.0.8" targetFramework="net471" />

<package id="Microsoft.jQuery.Unobtrusive.Validation" version="3.2.3" targetFramework="net471" />

<package id="Microsoft.Net.Compilers" version="2.4.0" targetFramework="net471" developmentDependency="true" />

<package id="Microsoft.Web.Infrastructure" version="1.0.0.0" targetFramework="net471" />

<package id="Modernizr" version="2.6.2" targetFramework="net471" />

<package id="Newtonsoft.Json" version="6.0.4" targetFramework="net471" />

<package id="Respond" version="1.2.0" targetFramework="net471" />

<package id="WebGrease" version="1.5.2" targetFramework="net471" />

</packages>

**WEB.CONFIG**

<?xml version="1.0" encoding="utf-8"?>

<!--

Para obter mais informações sobre como configurar seu aplicativo ASP.NET, visite

https://go.microsoft.com/fwlink/?LinkId=301880

-->

<configuration>

<configSections>

<section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" />

<!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 --></configSections>

<appSettings>

<add key="webpages:Version" value="3.0.0.0" />

<add key="webpages:Enabled" value="false" />

<add key="ClientValidationEnabled" value="true" />

<add key="UnobtrusiveJavaScriptEnabled" value="true" />

</appSettings>

<system.web>

<!--Restrição de acesso pelo form-->

<authentication mode="Forms">

<forms loginUrl="~/Account/Login" defaultUrl="~/" timeout="1" slidingExpiration="true"></forms>

</authentication>

<!--Restrição de acesso pelo nivel de acesso-->

<roleManager enabled="true" defaultProvider="MyProvider">

<providers>

<clear />

<add name="MyProvider" type="UniinfoAsp.Models.SiteRole"></add>

</providers>

</roleManager>

<compilation debug="true" targetFramework="4.7.1" />

<httpRuntime targetFramework="4.7.1" />

<httpModules>

<add name="ApplicationInsightsWebTracking" type="Microsoft.ApplicationInsights.Web.ApplicationInsightsHttpModule, Microsoft.AI.Web" />

</httpModules>

</system.web>

<runtime>

<assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">

<dependentAssembly>

<assemblyIdentity name="Newtonsoft.Json" culture="neutral" publicKeyToken="30ad4fe6b2a6aeed" />

<bindingRedirect oldVersion="0.0.0.0-6.0.0.0" newVersion="6.0.0.0" />

</dependentAssembly>

<dependentAssembly>

<assemblyIdentity name="System.Web.Optimization" publicKeyToken="31bf3856ad364e35" />

<bindingRedirect oldVersion="1.0.0.0-1.1.0.0" newVersion="1.1.0.0" />

</dependentAssembly>

<dependentAssembly>

<assemblyIdentity name="WebGrease" publicKeyToken="31bf3856ad364e35" />

<bindingRedirect oldVersion="0.0.0.0-1.5.2.14234" newVersion="1.5.2.14234" />

</dependentAssembly>

<dependentAssembly>

<assemblyIdentity name="System.Web.Helpers" publicKeyToken="31bf3856ad364e35" />

<bindingRedirect oldVersion="1.0.0.0-3.0.0.0" newVersion="3.0.0.0" />

</dependentAssembly>

<dependentAssembly>

<assemblyIdentity name="System.Web.WebPages" publicKeyToken="31bf3856ad364e35" />

<bindingRedirect oldVersion="1.0.0.0-3.0.0.0" newVersion="3.0.0.0" />

</dependentAssembly>

<dependentAssembly>

<assemblyIdentity name="System.Web.Mvc" publicKeyToken="31bf3856ad364e35" />

<bindingRedirect oldVersion="1.0.0.0-5.2.3.0" newVersion="5.2.3.0" />

</dependentAssembly>

</assemblyBinding>

</runtime>

<system.webServer>

<validation validateIntegratedModeConfiguration="false" />

<modules>

<remove name="ApplicationInsightsWebTracking" />

<add name="ApplicationInsightsWebTracking" type="Microsoft.ApplicationInsights.Web.ApplicationInsightsHttpModule, Microsoft.AI.Web" preCondition="managedHandler" />

</modules>

</system.webServer>

<system.codedom>

<compilers>

<compiler language="c#;cs;csharp" extension=".cs" type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CSharpCodeProvider, Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=1.0.8.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" warningLevel="4" compilerOptions="/langversion:default /nowarn:1659;1699;1701" />

<compiler language="vb;vbs;visualbasic;vbscript" extension=".vb" type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.VBCodeProvider, Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=1.0.8.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" warningLevel="4" compilerOptions="/langversion:default /nowarn:41008 /define:\_MYTYPE=\&quot;Web\&quot; /optionInfer+" />

</compilers>

</system.codedom>

<connectionStrings>

<add name="UnipEntities" connectionString="metadata=res://\*/Models.Uniinfo.csdl|res://\*/Models.Uniinfo.ssdl|res://\*/Models.Uniinfo.msl;provider=System.Data.SqlClient;provider connection string=&quot;data source=JORGE\SQLEXPRESS;initial catalog=Unip;user id=gestor;password=gestor;MultipleActiveResultSets=True;App=EntityFramework&quot;" providerName="System.Data.EntityClient" />

<!--<add name="UnipEntities" connectionString="metadata=res://\*/Models.Uniinfo.csdl|res://\*/Models.Uniinfo.ssdl|res://\*/Models.Uniinfo.msl;provider=System.Data.SqlClient;provider connection string=&quot;data source=LEANDRO-DEV\SQLEXPRESS;initial catalog=Unip;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework&quot;" providerName="System.Data.EntityClient" />-->

</connectionStrings>

<entityFramework>

<defaultConnectionFactory type="System.Data.Entity.Infrastructure.SqlConnectionFactory, EntityFramework" />

<providers>

<provider invariantName="System.Data.SqlClient" type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />

</providers>

</entityFramework>

</configuration>

<?xml version="1.0" encoding="utf-8"?>

<!-- Para obter mais informações sobre como usar a transformação do Web.config, visite https://go.microsoft.com/fwlink/?LinkId=301874 -->

<configuration xmlns:xdt="http://schemas.microsoft.com/XML-Document-Transform">

<!--

No exemplo abaixo, a transformação "SetAttributes" alterará o valor de

"connectionString" para usar "ReleaseSQLServer" apenas quando o localizador "Match"

encontrar um atributo "name" que tenha um valor de "MyDB".

<connectionStrings>

<add name="MyDB"

connectionString="Data Source=ReleaseSQLServer;Initial Catalog=MyReleaseDB;Integrated Security=True"

xdt:Transform="SetAttributes" xdt:Locator="Match(name)"/>

</connectionStrings>

-->

<system.web>

<!--

No exemplo abaixo, a transformação "Replace" vai substituir toda a seção

<customErrors> do seu arquivo Web.confige.

Observe que como há somente uma seção customErrors no nó

<system.web>, não há necessidade de usar o atributo "xdt:Locator".

<customErrors defaultRedirect="GenericError.htm"

mode="RemoteOnly" xdt:Transform="Replace">

<error statusCode="500" redirect="InternalError.htm"/>

</customErrors>

-->

</system.web>

</configuration>

<?xml version="1.0" encoding="utf-8"?>

<!-- Para obter mais informações sobre como usar a transformação do Web.config, visite https://go.microsoft.com/fwlink/?LinkId=301874 -->

<configuration xmlns:xdt="http://schemas.microsoft.com/XML-Document-Transform">

<!--

No exemplo abaixo, a transformação "SetAttributes" alterará o valor de

"connectionString" para usar "ReleaseSQLServer" apenas quando o localizador "Match"

encontrar um atributo "name" que tenha um valor de "MyDB".

<connectionStrings>

<add name="MyDB"

connectionString="Data Source=ReleaseSQLServer;Initial Catalog=MyReleaseDB;Integrated Security=True"

xdt:Transform="SetAttributes" xdt:Locator="Match(name)"/>

</connectionStrings>

-->

<system.web>

<compilation xdt:Transform="RemoveAttributes(debug)" />

<!--

No exemplo abaixo, a transformação "Replace" vai substituir toda a seção

<customErrors> do seu arquivo Web.config.

Observe que como há somente uma seção customErrors no nó

<system.web>, não há necessidade de usar o atributo "xdt:Locator".

<customErrors defaultRedirect="GenericError.htm"

mode="RemoteOnly" xdt:Transform="Replace">

<error statusCode="500" redirect="InternalError.htm"/>

</customErrors>

-->

</system.web>

</configuration>

**WebService**

using System;

using System.Collections.Generic;

using System.Web.Services;

using UniinfoAsp.DAL;

using UniinfoAsp.Models;

/// <summary>

/// Descrição resumida de WebService

/// </summary>

//[WebService(Namespace = "http://tempuri.org/")]

[WebService(Namespace = "http://microsoft.com/webservices/")]

[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]

// Para permitir que esse serviço da web seja chamado a partir do script, usando ASP.NET AJAX, remova os comentários da linha a seguir.

// [System.Web.Script.Services.ScriptService]

public class WebService : System.Web.Services.WebService

{

chamadoDAO dao = new chamadoDAO();

public WebService()

{

//Remova os comentários da linha a seguir se usar componentes designados

InitializeComponent();

}

private void InitializeComponent()

{

throw new NotImplementedException();

}

[WebMethod]

public string HelloWorld()

{

return "Olá, Mundo";

}

[WebMethod]

public List<Con\_Chamado> ConsultarChamado()

{

return dao.Consultar();

}

[WebMethod]

public List<Chamado> Consultar2()

{

return dao.Consultar2();

}

[WebMethod]

public string verificanivel(string loginwpf, string senhawpf)

{

return dao.verificaLogin(loginwpf, senhawpf);

}

}

**Home.Aspx**

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Home.aspx.cs" Inherits="Home" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<title></title>

</head>

<body>

<a href="WebService.asmx">web</a>

<form id="form1" runat="server">

<div>

</div>

</form>

</body>

</html>

**PACKAGES.CONFIG**

<?xml version="1.0" encoding="utf-8"?>

<packages>

<package id="Microsoft.CodeDom.Providers.DotNetCompilerPlatform" version="2.0.0" targetFramework="net471" />

</packages>

**WEB.CONFIG**

<?xml version="1.0" encoding="utf-8"?>

<!--

Para obter mais informações sobre como configurar seu aplicativo ASP.NET, visite

https://go.microsoft.com/fwlink/?LinkId=169433

-->

<configuration>

<system.web>

<compilation debug="true" targetFramework="4.7.1"/>

<httpRuntime targetFramework="4.7.1"/>

</system.web>

<system.codedom>

<compilers>

<compiler language="c#;cs;csharp" extension=".cs"

type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CSharpCodeProvider, Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"

warningLevel="4" compilerOptions="/langversion:default /nowarn:1659;1699;1701"/>

<compiler language="vb;vbs;visualbasic;vbscript" extension=".vb"

type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.VBCodeProvider, Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"

warningLevel="4" compilerOptions="/langversion:default /nowarn:41008 /define:\_MYTYPE=\&quot;Web\&quot; /optionInfer+"/>

</compilers>

</system.codedom>

<connectionStrings>

<add name="UnipEntities" connectionString="metadata=res://\*/Models.Uniinfo.csdl|res://\*/Models.Uniinfo.ssdl|res://\*/Models.Uniinfo.msl;provider=System.Data.SqlClient;provider connection string=&quot;data source=JORGE\SQLEXPRESS;initial catalog=Unip;user id=gestor;password=gestor;MultipleActiveResultSets=True;App=EntityFramework&quot;" providerName="System.Data.EntityClient" />

<!--<add name="UnipEntities" connectionString="metadata=res://\*/Models.Uniinfo.csdl|res://\*/Models.Uniinfo.ssdl|res://\*/Models.Uniinfo.msl;provider=System.Data.SqlClient;provider connection string=&quot;data source=LEANDRO-DEV\SQLEXPRESS;initial catalog=Unip;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework&quot;" providerName="System.Data.EntityClient" />-->

</connectionStrings>

</configuration>

<?xml version="1.0" encoding="utf-8"?>

<!-- Para obter mais informações sobre como usar a transformação do web.config, visite https://go.microsoft.com/fwlink/?LinkId=125889 -->

<configuration xmlns:xdt="http://schemas.microsoft.com/XML-Document-Transform">

<!--

No exemplo abaixo, a transformação de "SetAttributes" alterará o valor de

"connectionString" para usar "ReleaseSQLServer" somente quando o localizador "Match"

encontrar um atributo "name" que tem o valor de "MyDB".

<connectionStrings>

<add name="MyDB"

connectionString="Data Source=ReleaseSQLServer;Initial Catalog=MyReleaseDB;Integrated Security=True"

xdt:Transform="SetAttributes" xdt:Locator="Match(name)"/>

</connectionStrings>

-->

<system.web>

<compilation xdt:Transform="RemoveAttributes(debug)" />

<!--

No exemplo a seguir, a transformação "Replace" substituirá toda a

seção <customErrors> do arquivo web.config.

Observe que como existe apenas uma seção customErrors sob o nó

<system.web>, não há necessidade de usar o atributo "xdt:Locator".

<customErrors defaultRedirect="GenericError.htm"

mode="RemoteOnly" xdt:Transform="Replace">

<error statusCode="500" redirect="InternalError.htm"/>

</customErrors>

>

-->

</system.web>

</configuration>

**WebService.asmx**

<%@ WebService Language="C#" CodeBehind="WebService/App\_Code/WebService.cs" Class="WebService"%>

# APÊNDICE B - CÓDIGO FONTE WPF

**Classe do WPF Inicio.xaml**

using System.Windows;

using System.Windows.Controls;

using Uniinfo\_Desk.ServiceReference;

namespace Uniinfo\_Desk.Apresentação

{

public partial class Inicio : Window

{

public Inicio()

{

InitializeComponent();

}

//Evento do Botão Atualizar

public void btnLogar\_Click\_1(object sender, RoutedEventArgs e)

{

Grid\_Loaded(sender, e);

}

//Evento do data Grid, para carregar os dados.

public void Grid\_Loaded(object sender, RoutedEventArgs e)

{

try

{

WebServiceSoapClient obj = new WebServiceSoapClient();

grid.ItemsSource = obj.ConsultarChamado();

}

catch (System.Exception)

{

MessageBox.Show("Falha na Conexão com o Web Service");

}

}

//Evento do data Grid, para configurar colunas

private void grid\_AutoGeneratingColumn(object sender, System.Windows.Controls.DataGridAutoGeneratingColumnEventArgs e)

{

//Ocultando uma coluna

if (e.PropertyName == "ExtensionData")

{

e.Column = null;

}

//Formatando os dados de uma coluna

if (e.PropertyName == "dataChamado")

{

(e.Column as DataGridTextColumn).Binding.StringFormat = "dd/MM/yyyy";

e.Column.Header = "Data de Abertura";

}

//Formatando o Header das colunas

if (e.PropertyName.StartsWith("descricao"))

{

e.Column.Header = "Descrição";

}

if (e.PropertyName.StartsWith("statusAtendimento"))

{

e.Column.Header = "Status do Atendimento";

}

if (e.PropertyName.StartsWith("nome"))

{

e.Column.Header = "Nome";

}

}

//Evento do Botão do Loggout

private void btnLoggout\_Click(object sender, RoutedEventArgs e)

{

MainWindow Login = new MainWindow();

mnConsulta.Close();

Login.mnaLogin.ShowDialog();

}

}

}

**Classe do WPF Login.xaml**

using System;

using System.Threading;

using System.Windows;

using Uniinfo\_Desk.Apresentação;

using Uniinfo\_Desk.ServiceReference;

namespace Uniinfo\_Desk

{

public partial class MainWindow : Window

{

Inicio Inicio = new Inicio();

SplashScreen splash = new SplashScreen("Apresentação/home.png");

public MainWindow()

{

InitializeComponent();

}

//Evento do Botão de Login

private void btnLogar\_Click\_1(object sender, RoutedEventArgs e)

{

//Tenta fazer a conexão no WebService

try

{

//Chamando o metódo de Login do WebService

WebServiceSoapClient obj = new WebServiceSoapClient();

string Status = obj.verificanivel(this.txbUsuário.Text, this.psdSenha.Password.ToString());

//Validando o Login e Senha

if (txbUsuário.Text == "" && psdSenha.Password == "") { txbErros.Text = "Digite seu Usuário e sua Senha"; }

else if (txbUsuário.Text == "") { txbErros.Text = "Digite seu Usuário"; }

else if (psdSenha.Password == "") { txbErros.Text = "Digite sua Senha"; }

else if (Status == "true")

{

mnaLogin.Close();

//Efeito de abertura com Imagem

SplashScreen splash = new SplashScreen("Apresentação/Imagens/Inicial.png");

splash.Show(false);

Thread.Sleep(3000);

splash.Close(new TimeSpan(0, 0, 1));

Inicio.mnConsulta.Show();

Inicio.Grid\_Loaded(sender, e);

}

else { txbErros.Text = "Usuário ou Senha incorreto"; }

}

//Caso de erro na Conexão

catch (System.Exception)

{

txbErros.Text= "Falha na conexão com o servidor de Login";

}

}

private void txbSenha\_PasswordChanged(object sender, RoutedEventArgs e)

{

}

}

}

**APP.xaml**

<Application x:Class="Uniinfo\_Desk.App"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="clr-namespace:Uniinfo\_Desk"

StartupUri="Apresentação/Login.xaml">

<Application.Resources>

<ResourceDictionary>

<ResourceDictionary.MergedDictionaries>

<ResourceDictionary Source="pack://application:,,,/MaterialDesignThemes.Wpf;component/Themes/MaterialDesignTheme.Light.xaml" />

<ResourceDictionary Source="pack://application:,,,/MaterialDesignThemes.Wpf;component/Themes/MaterialDesignTheme.Defaults.xaml" />

<ResourceDictionary Source="pack://application:,,,/MaterialDesignColors;component/Themes/Recommended/Primary/MaterialDesignColor.Blue.xaml" />

<ResourceDictionary Source="pack://application:,,,/MaterialDesignColors;component/Themes/Recommended/Accent/MaterialDesignColor.Lime.xaml" />

</ResourceDictionary.MergedDictionaries>

</ResourceDictionary>

</Application.Resources>

</Application>

**Packages.config**

<?xml version="1.0" encoding="utf-8"?>

<packages>

<package id="AngleSharp" version="0.9.9" targetFramework="net461" />

<package id="MaterialDesignColors" version="1.1.3" targetFramework="net461" />

<package id="MaterialDesignThemes" version="2.5.0.1205" targetFramework="net461" />

<package id="MaterialSkin" version="0.2.1" targetFramework="net461" />

<package id="XamlCSS" version="2.1.0" targetFramework="net461" />

<package id="XamlCSS.WPF" version="2.1.0" targetFramework="net461" />

</packages>

**App.config**

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<startup>

<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6.1" />

</startup>

<system.serviceModel>

<bindings>

<basicHttpBinding>

<binding name="WebServiceSoap" />

</basicHttpBinding>

</bindings>

<client>

<endpoint address="http://localhost:64848/WebSevice/WebService.asmx"

binding="basicHttpBinding" bindingConfiguration="WebServiceSoap"

contract="ServiceReference.WebServiceSoap" name="WebServiceSoap" />

</client>

</system.serviceModel>

</configuration>