**UNIVERSIDADE PAULISTA**

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**EMERSON GASPARINI JUNIOR - N19044-2**

**HEYTTOR FLEMMING ANTUNES SILVA – D3402F-2**

**IGOR MENDES MEDEIROS – N117DB-2**

**JULLIANN ANDRÉ RAMOS – N139JF-2**

**MURILO DE ALMEIDA LIMA – D223JC-6**

**SISTEMA DE HELP-DESK**

**EVOLUTION**

**SOROCABA-SP**

**2018**

**EMERSON GASPARINI JUNIOR - N19044-2**

**HEYTTOR FLEMMING ANTUNES SILVA – D3402F-2**

**IGOR MENDES MEDEIROS – N117DB-2**

**JULLIANN ANDRÉ RAMOS – N139JF-2**

**MURILO DE ALMEIDA LIMA – D223JC-6**

**SISTEMA DE HELP-DESK**

**Trabalho de conclusão do semestre na forma de projeto integrado multidisciplinar (PIM) apresentado à Universidade Paulista - UNIP**

**Orientador: Todos os Professores do Semestre**

**SOROCABA-SP**

**2018**

**EMERSON GASPARINI JUNIOR - N19044-2**

**HEYTTOR FLEMMING ANTUNES SILVA - D3402F-2**

**IGOR MENDES MEDEIROS - N117DB-2**

**JULLIANN ANDRÉ RAMOS - N139JF-2**

**MURILO DE ALMEIDA LIMA - D223JC-6**

**UNIP-MAP**

**APLICATIVO DE LOCALIZAÇÃO**

**Trabalho de conclusão do semestre na forma de projeto integrado multidisciplinar (PIM) apresentado à Universidade Paulista - UNIP**

**Orientador: Todos os Professores do Semestre**

**Nota:\_\_\_\_\_\_\_\_\_**

**Richardson Luz**

**Universidade Paulista**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_/\_\_\_**

**Waldir Silva**

**Universidade Paulista**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_/\_\_\_**

**Elizeu Elieber**

**Universidade Paulista**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_/\_\_\_**

**Reverdan Sparinger**

**Universidade Paulista**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_/\_\_\_**

**RESUMO**

Este projeto apresenta o sistema de help-desk Evolution, que foi solicitado à empresa M.I.J.H Systems. O sistema será disponibilizado para uma empresa que prestara serviços de suporte técnico para outras empresas através do sistema de help-desk Evolution. Este sistema foi desenvolvido usando as linguagens C# e ASP.NET, com um banco de dados em SQLServer, que servira para empresas que contratarem o serviço de suporte técnico terem suporte sobre os produtos da empresa. Este documento contém toda a parte desenvolvimento do sistema, começando pela análise de requisitos, recursos e custos, fluxogramas e os layouts de tela, passando pelos códigos e banco de dados ate os testes feitos no software.

**ABSTRACT**

**INDICE DE ILUSTRAÇÕES**

Figura 1-Tabela de custos 12

Figura 2-Tabela de salários 12

Figura 3-Tabela de linhas de base 12

Figura 4-Casos de uso 13

Figura 5-Fluxograma do modulo web 14

Figura 6-Fluxograma do modulo desktop 15

Figura 7-Diagrama de atividade do sistema 16

Figura 8-Diagrama de classes 17

Figura 9-Diagrama de sequência do desktop 18

Figura 10-Tela de login do modulo desktop 19

Figura 11-Tela home do desktop 19

Figura 12-Tela de cadastrar usuário 20

Figura 13-Tela de editar usuário 20

Figura 14-Tela de editar cliente 21

Figura 15-Tela de cadastrar funcionário 22

Figura 16-Tela de editar funcionário 22

Figura 17-Tela de cadastrar produto 23

Figura 18-Tela de editar produto 23

Figura 19-Tela de cadastrar e visualizar atendimento 24

Figura 20-Diagrama de sequência do modulo web 25

Figura 21-Tela do login do modulo web 25

Figura 22-Tela do cliente na web 26

Figura 23-Tela do funcionário na web 26

Figura 24-Modelo conceitual 28

Figura 25-Modelo logico 28

Figura 26-APP.Config 29

Figura 27-WEB.Config 30

**SUMARIO**

[1.0 Introdução 10](#_Toc530498544)

[2.0 Projeto help desk Evolution 11](#_Toc530498545)

[2.1 Objetivo 11](#_Toc530498546)

[2.2 Suposições e Restrições 11](#_Toc530498547)

[2.3 Recursos 12](#_Toc530498548)

[2.4 custos 12](#_Toc530498549)

[2.5 Datas das Fases e Interações das Linhas Base 12](#_Toc530498550)

[2.6 casos de uso 13](#_Toc530498551)

[3.0 Sistema help desk Evolution 14](#_Toc530498552)

[3.0.1 Fluxogramas 14](#_Toc530498553)

[3.0.2 Diagrama de atividades 16](#_Toc530498554)

[3.0.3 Diagrama de classes 17](#_Toc530498555)

[3.1 Desktop 17](#_Toc530498556)

[3.1.1 Diagrama de sequência 18](#_Toc530498557)

[3.1.2 Layout de tela 19](#_Toc530498558)

[3.2 Web 24](#_Toc530498559)

[3.2.1 Diagrama de sequência 25](#_Toc530498560)

[3.2.2 Layouts de telas 25](#_Toc530498561)

[4.0 Desenvolvimento do projeto 27](#_Toc530498562)

[4.1 Código da parte desktop 27](#_Toc530498563)

[4.2 Código da parte web 27](#_Toc530498564)

[5.0 Banco de dados 28](#_Toc530498565)

[5.1 Modelo Conceitual 28](#_Toc530498566)

[5.2 Modelo Logico 28](#_Toc530498567)

[5.3 Modelo Físico 28](#_Toc530498568)

[5.4 Conexões com o banco de dados 29](#_Toc530498569)

[5.4.1 Conexão desktop 29](#_Toc530498570)

[5.4.2 Conexão Web 30](#_Toc530498571)

[6.0 Requisitos de uso e desenvolvimento 31](#_Toc530498572)

[7.0 Parametros de Qualidade 32](#_Toc530498573)

[7.1 Funcionalidades 32](#_Toc530498574)

[7.1.2 Adequação 32](#_Toc530498575)

[7.1.3 Acurácia 32](#_Toc530498576)

[7.1.4 Interoperabilidade 32](#_Toc530498577)

[7.1.5 Segurança 32](#_Toc530498578)

[7.2 Confiabilidade 32](#_Toc530498579)

[7.2.1 Maturidade 32](#_Toc530498580)

[7.2.2 Tolerância a falhas 32](#_Toc530498581)

[7.2.3 Recuperabilidade 32](#_Toc530498582)

[7.3 Usabilidade 33](#_Toc530498583)

[7.3.1 Inteligibilidade 33](#_Toc530498584)

[7.3.2 Apreensibilidade 33](#_Toc530498585)

[7.4 Manutenibilidade 33](#_Toc530498586)

[7.4.1 Analisabilidade 33](#_Toc530498587)

[7.4.2 Modificabilidade 33](#_Toc530498588)

[7.4.3 Estabilidade 33](#_Toc530498589)

[8.0 Testes 34](#_Toc530498590)

[8.1 Testes do modulo desktop 34](#_Toc530498591)

[9.0 Matérias do Semestre 41](#_Toc530498592)

[9.1 Programação Orientada a Objetos 2 41](#_Toc530498593)

[9.2 Tópicos especiais de programação orientada a objetos 41](#_Toc530498594)

[9.3 Desenvolvimento de Software para Internet 41](#_Toc530498595)

[9.4 Projeto Orientado a Objetos 41](#_Toc530498596)

[9.5 Gerenciamento de projeto de software 41](#_Toc530498597)

[10 Conclusão 42](#_Toc530498598)

[Referencias 43](#_Toc530498599)

# 1.0 Introdução

Neste semestre o tema do projeto integrado multidisciplinar, foi para que os alunos do terceiro e quarto semestre criassem um sistema de help-desk, o sistema será desenvolvido para uma empresa de prestação de serviços, que prestara serviços de suporte técnico para outras empresas através do sistema. O sistema Evolution, possui os recursos necessários para a administração de seus funcionários, clientes, dos chamados realizados e os produtos da empresa contratante. Através do sistema Evolution as empresas contratantes podem ter acesso a suporte para os produtos de sua empresa e podem contar com técnicos especializados para o serviço.

# 2.0 Projeto help desk Evolution

## 2.1 Objetivo

O objetivo do Sistema Evolution é possibilitar aos gestores terem um ambiente organizado e padronizado que permita a empresa identificar possíveis problemas no relacionamento de atendimento ao cliente. Analisado os problemas através do atendimento a distância focando no tempo para solução, as etapas realizadas até a solução, visando o grau de satisfação do cliente, e verificar meios de melhor atendimento e tomada de decisão.

Oferecendo um ambiente moderno para o profissional e para o cliente, podendo assim ampliar o relacionamento entre cliente e empresa.

Podendo também realizar uma avaliação dos colaboradores e possíveis problemas.

Além de arquivar informação dos atendimentos, estando disponíveis para a visualização.

## 2.2 Suposições e Restrições

Em seu desenvolvimento, consideramos alguns erros que os usuários poderiam cometer ao introduzir as informações, o sistema possui 3 níveis de usuário, sendo eles, coordenador, funcionário e cliente cada um com suas restrições. Para que um cliente possa abrir um atendimento ele deverá ter um login. O login poderá ter apenas um papel (Cliente ou Funcionário). Os usuários que possuírem o papel de Cliente terão acesso apenas a página web. Os usuários que possuírem o papel de Funcionário poderão ter o nível Coordenador ou Funcionário e terão acesso ao módulo desktop e página web. Quando o login tem o papel de funcionário, não poderá haver logins diferentes para o mesmo funcionário. Já para o cliente, poderá haver nomes de logins diferentes para o mesmo cliente.

## 2.3 Recursos

Para a produção do sistema Evolution foram usados softwares para montagem dos códigos, elaboração das interfaces e criação e administração do banco de dados, tais foram eles respectivamente, Visual Studio, Photoshop e MySQL Server 2012

Para a implementação do sistema foi primeiramente preciso a criação de um ambiente qualificado para a realização de tais tarefas, após a criação dos ambientes foi feito a aquisição dos equipamentos necessários e então pôr fim a implementação do servidor adequado para o funcionamento do sistema.

## 2.4 custos

Figura 1-Tabela de custos



Fonte:M.I.J.H Systens

Figura 2-Tabela de salários



Fonte:M.I.J.H Systens

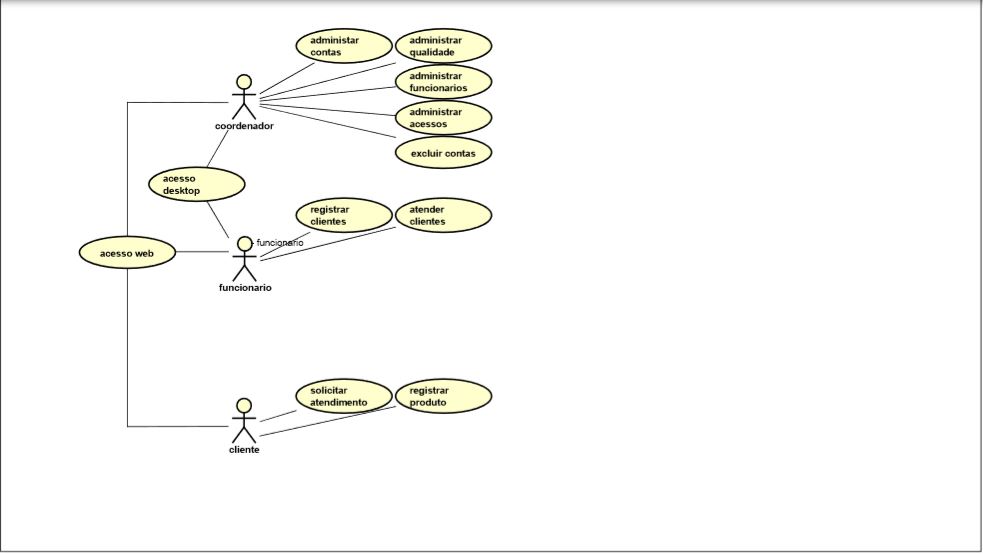
## 2.5 Datas das Fases e Interações das Linhas Base

Figura 3-Tabela de linhas de base

|  |  |  |  |
| --- | --- | --- | --- |
| Datas das Fases e Iterações das Linhas Base | | | |
| Fase RUP | Iteração RUP | Linha Base | Data 05/06/2018  Alvo 15/07/2018 |
| Inception | Iniciação | Funcional | Data 25/07/2018 |
| Elaboration | Protótipo de  Arquitetura | Projeto | Data 15/08/2018  Alvo 01/09/2018 |
| Contruction | Release de Protótipo | Produto | Data 09/11/2018 |
| Transaction | Release Final | Produto | Data 13/11/2018 |

## 2.6 casos de uso

Figura 4-Casos de uso



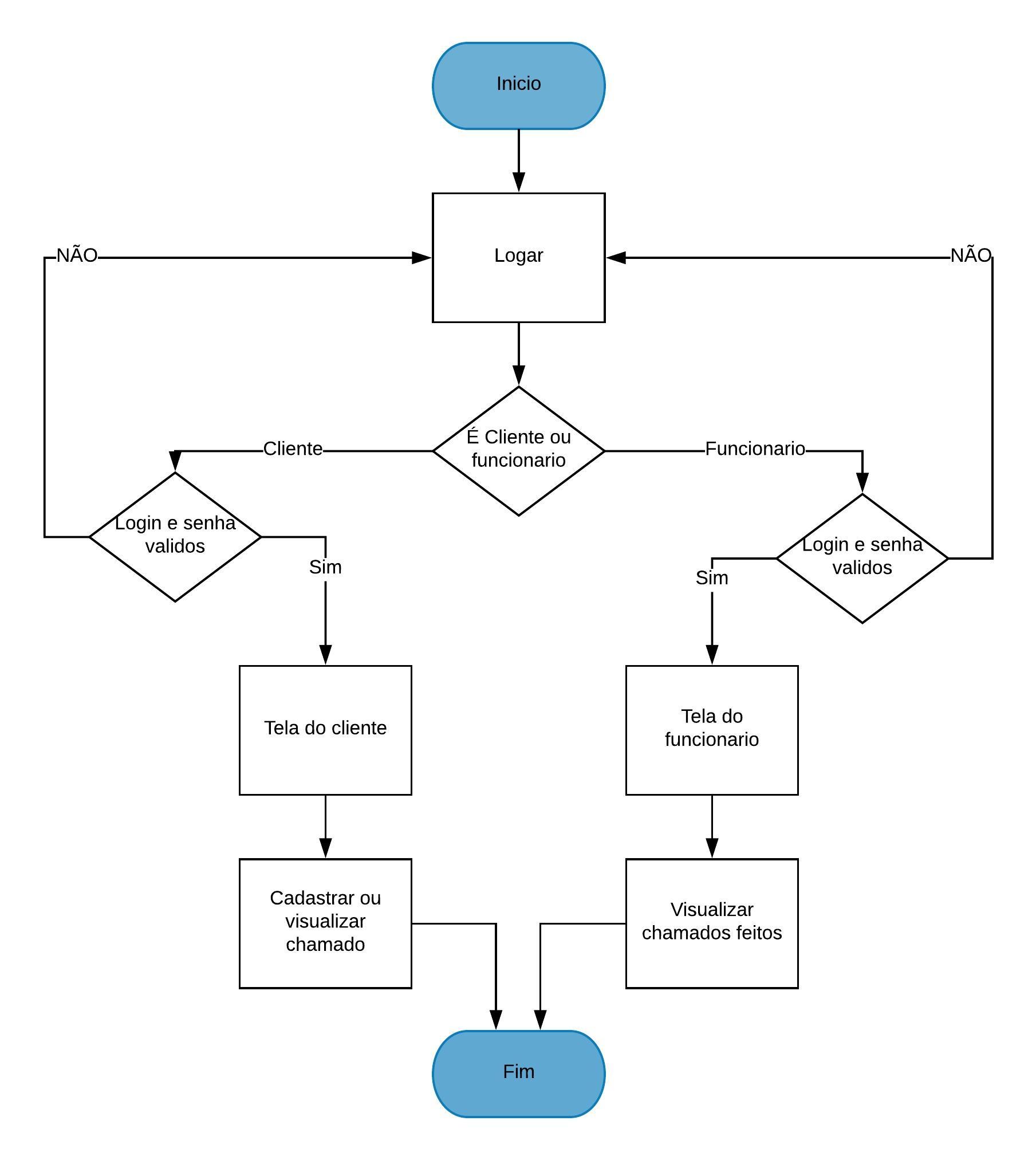
Fonte:M.I.J.H Systens

# 3.0 Sistema help desk Evolution

Este sistema foi desenvolvido em linguagens C# e ASP.NET, a linguagem C# foi utilizada no desenvolvimento da parte desktop do sistema que será a parte de administração onde somente os técnicos da empresa de suporte utilizarão, utilizando ASP.NET para a parte web do sistema que servira para uso do cliente, realizando chamadas para entrar em contato com o suporte técnico.

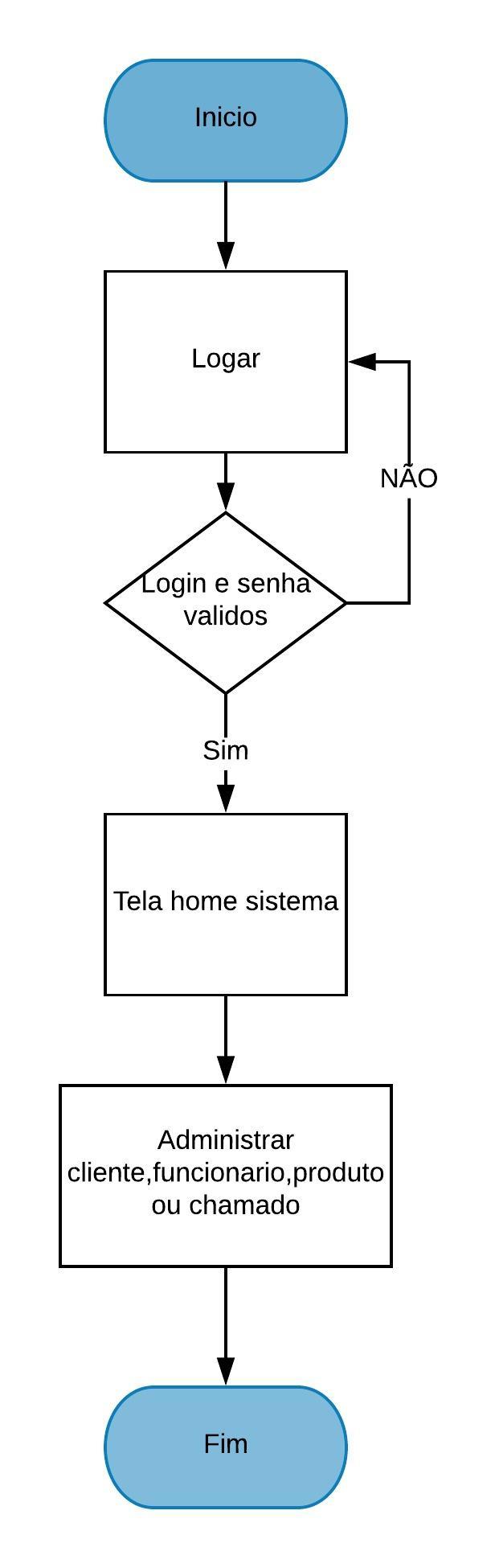
## 3.0.1 Fluxogramas

Figura 5-Fluxograma do modulo web



Fonte:M.I.J.H Systens

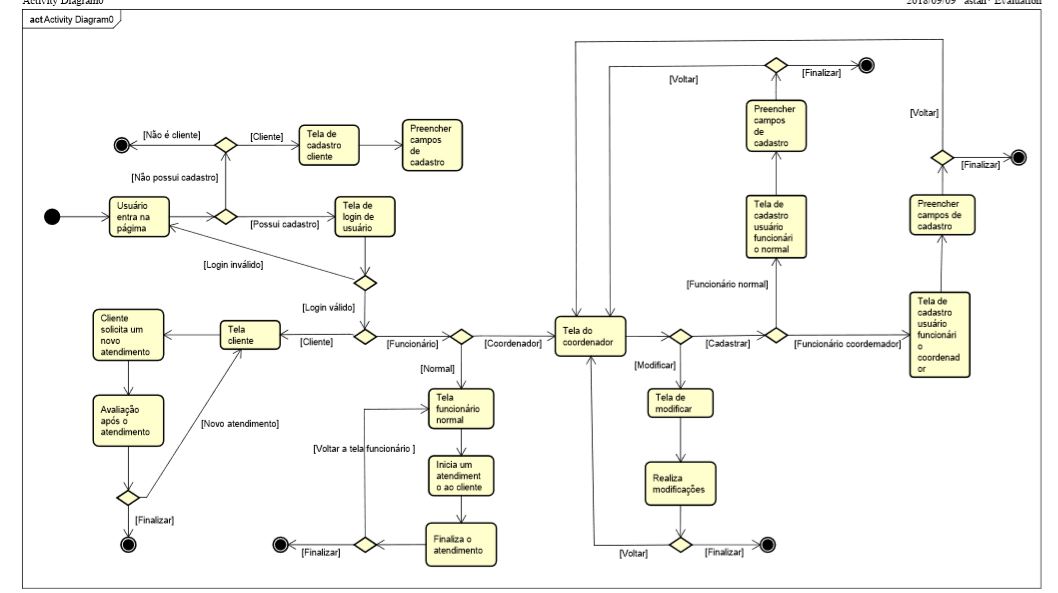
Figura 6-Fluxograma do modulo desktop



Fonte:M.I.J.H Systens

## 3.0.2 Diagrama de atividades

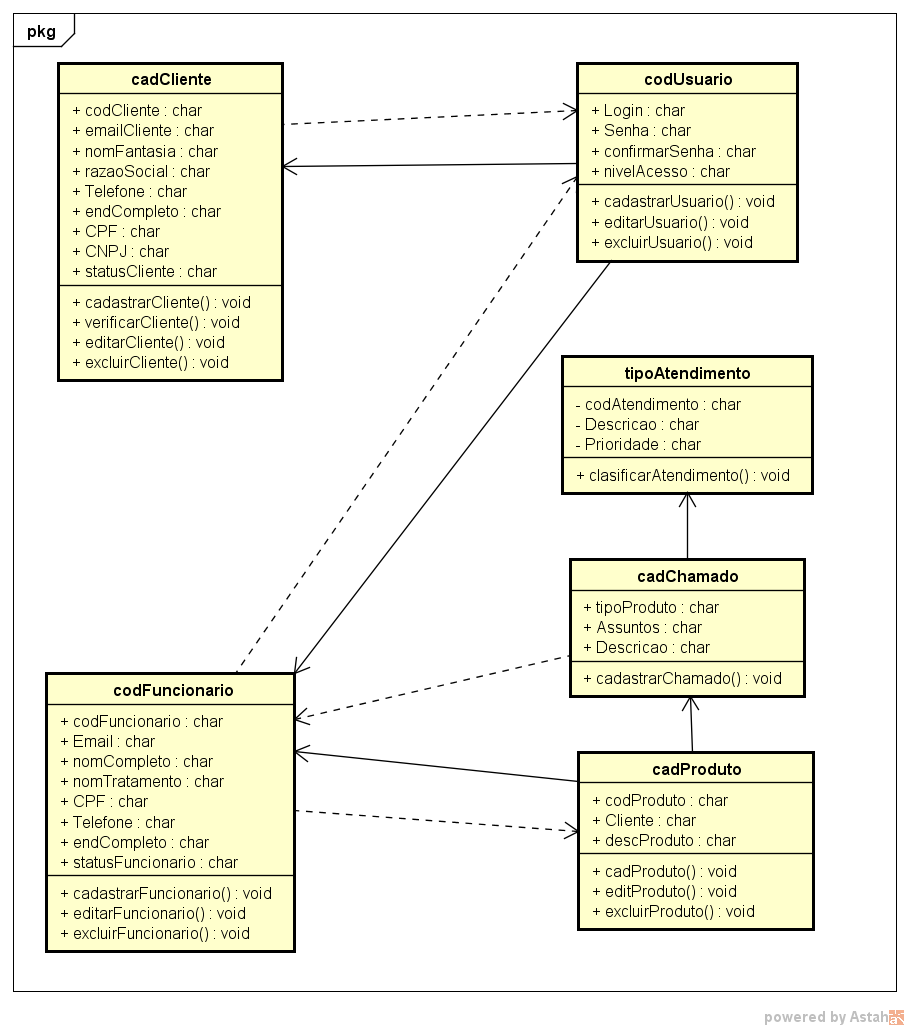
Figura 7-Diagrama de atividade do sistema



Fonte:M.I.J.H Systens

## 3.0.3 Diagrama de classes

Figura 8-Diagrama de classes



Fonte:M.I.J.H Systens

## 3.1 Desktop

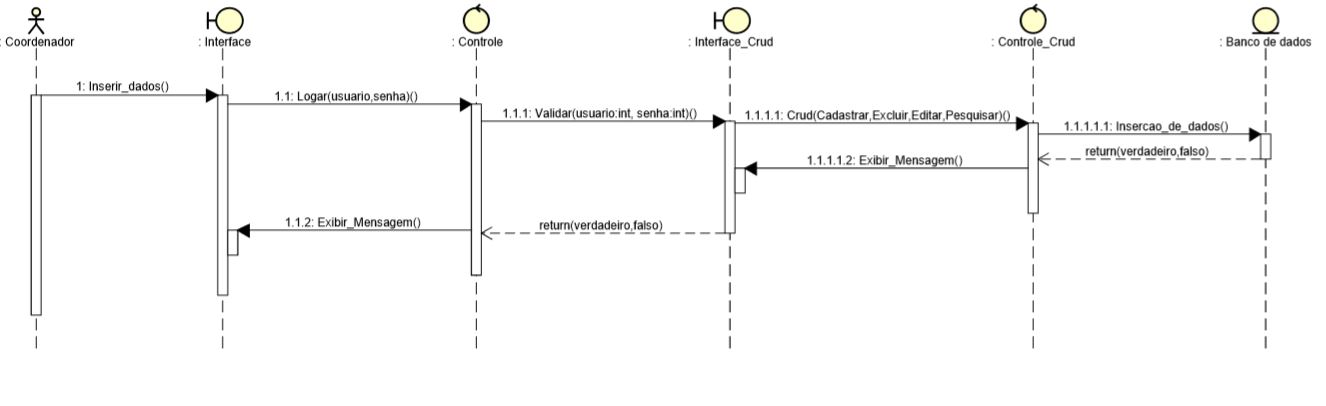
Este sistema foi desenvolvido em C# tanto na parte backend e frontend, utilizando o framework entity onde ele mapeia as classes e tabelas do banco, criando interação entre si

Foi utilizada a arquitetura MVC (model, view, controller), que divide a representação da informação do usuário com ele. É normalmente usado para o desenvolvimento de interface de usuário que divide uma aplicação em três parte interconectadas. Isto é feito para separar representações de informação internas dos modos como a informação é apresentada para e aceita pelo usuário.

A parte do sistema que ficara no desktop dos profissionais da empresa que prestara o serviço de suporte técnico. Onde somente o administrador e seus funcionários irão manuseá-los, este modulo terá a função de administrar o sistema, onde o administrador terá controle total sobre contas de clientes e funcionários, e o técnico terá controle sobre as contas dos clientes.

## 3.1.1 Diagrama de sequência

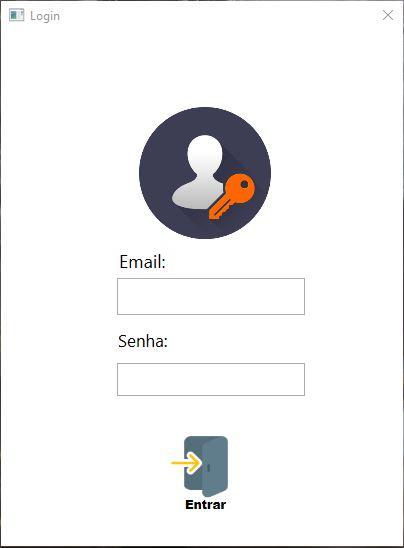
Figura 9-Diagrama de sequência do desktop



Fonte:M.I.J.H Systens

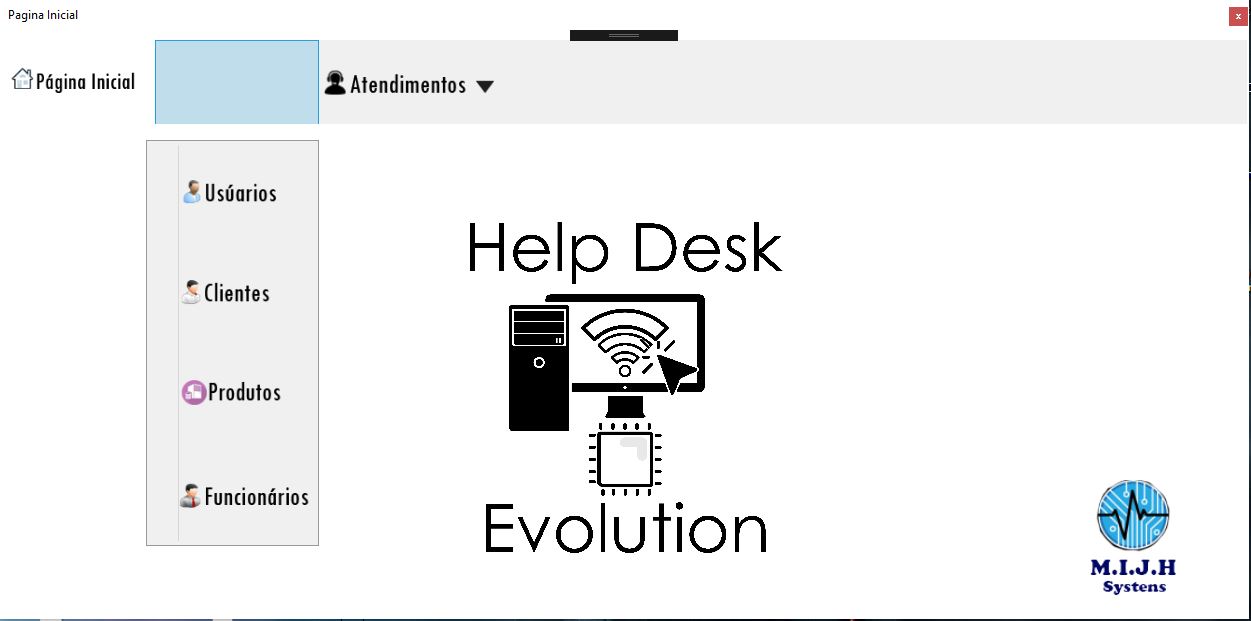
## 3.1.2 Layout de tela

Figura 10-Tela de login do modulo desktop



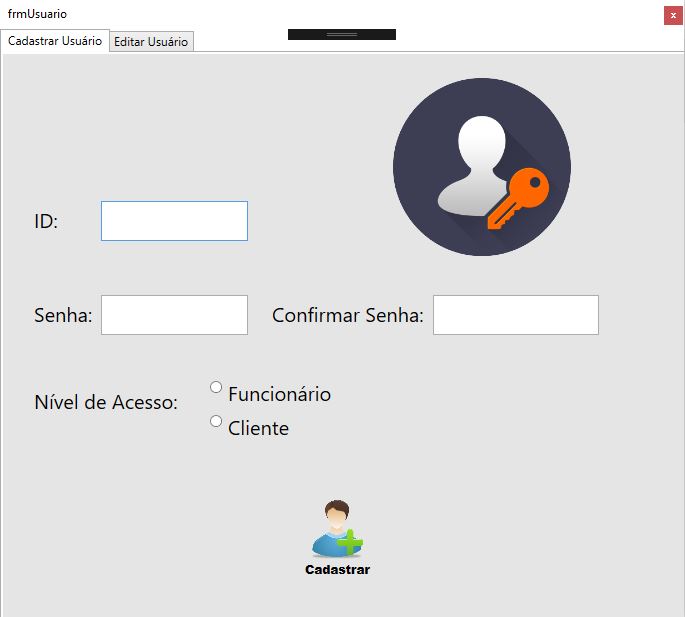
Fonte:M.I.J.H Systens

Figura 11-Tela home do desktop



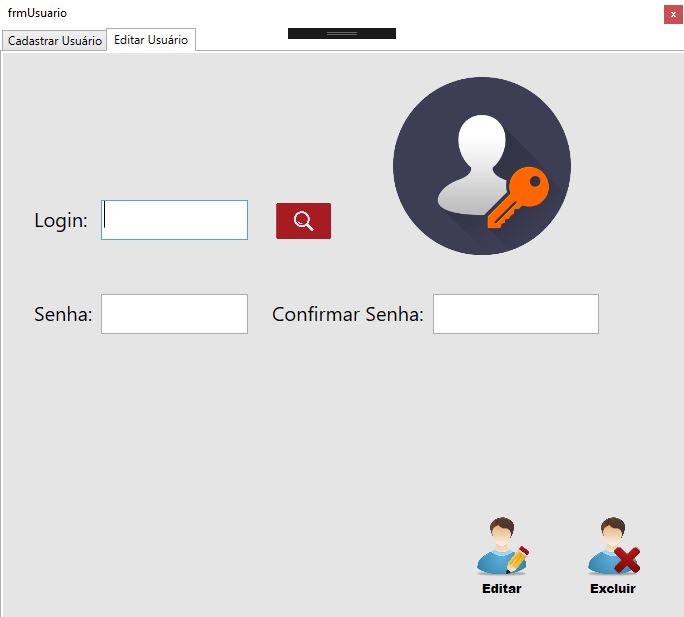
Fonte:M.I.J.H Systens

Figura 12-Tela de cadastrar usuário

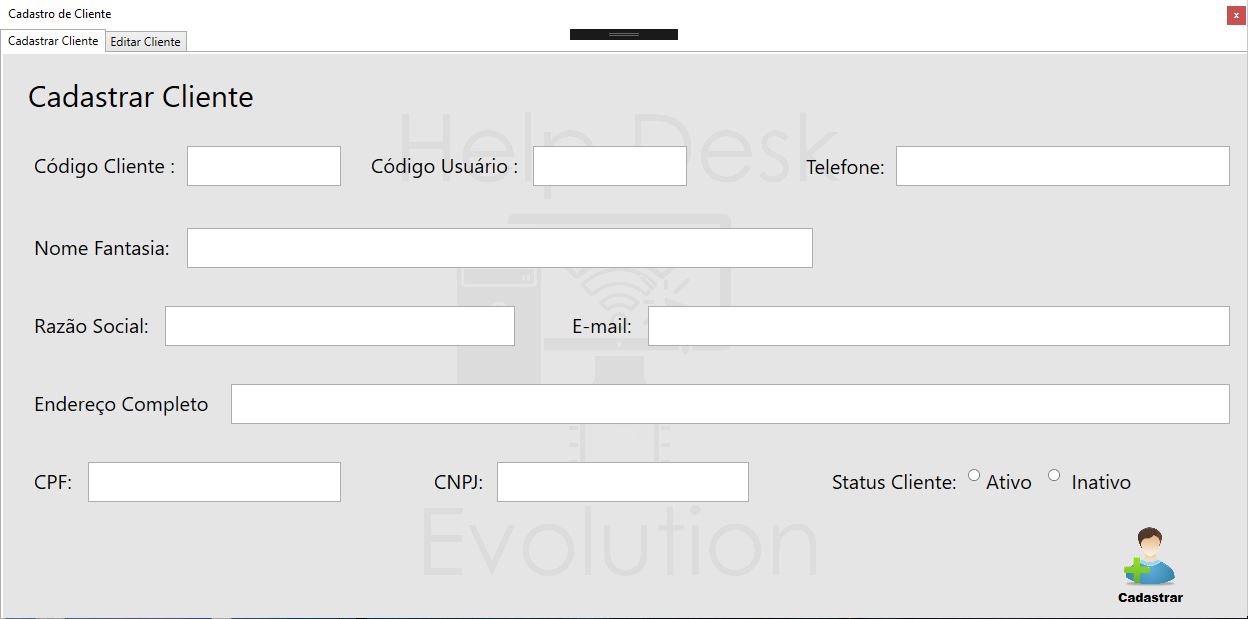


Fonte:M.I.J.H Systens

Figura 13-Tela de editar usuário

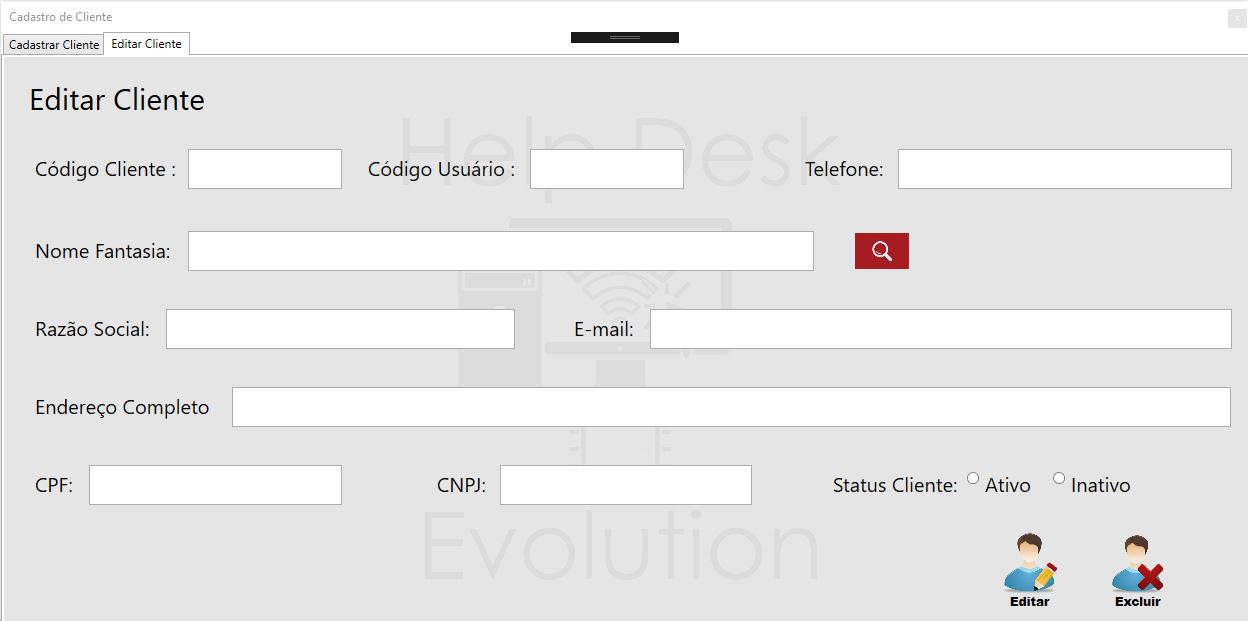


Fonte:M.I.J.H Systens



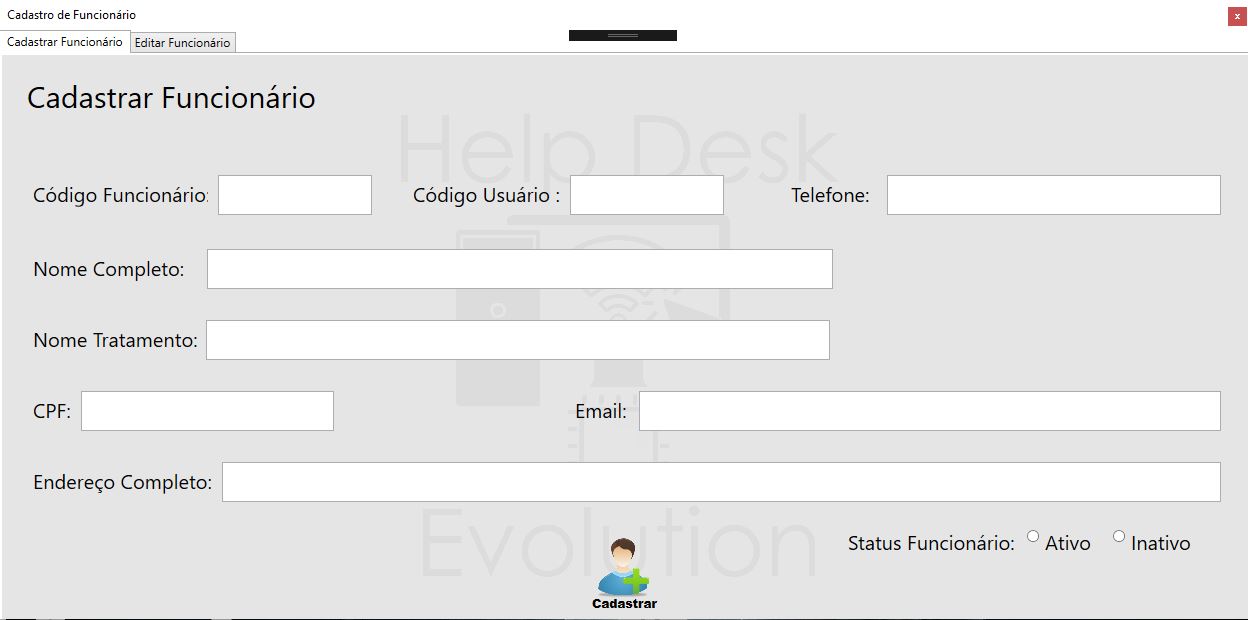
Fonte:M.I.J.H Systens

Figura 14-Tela de editar cliente



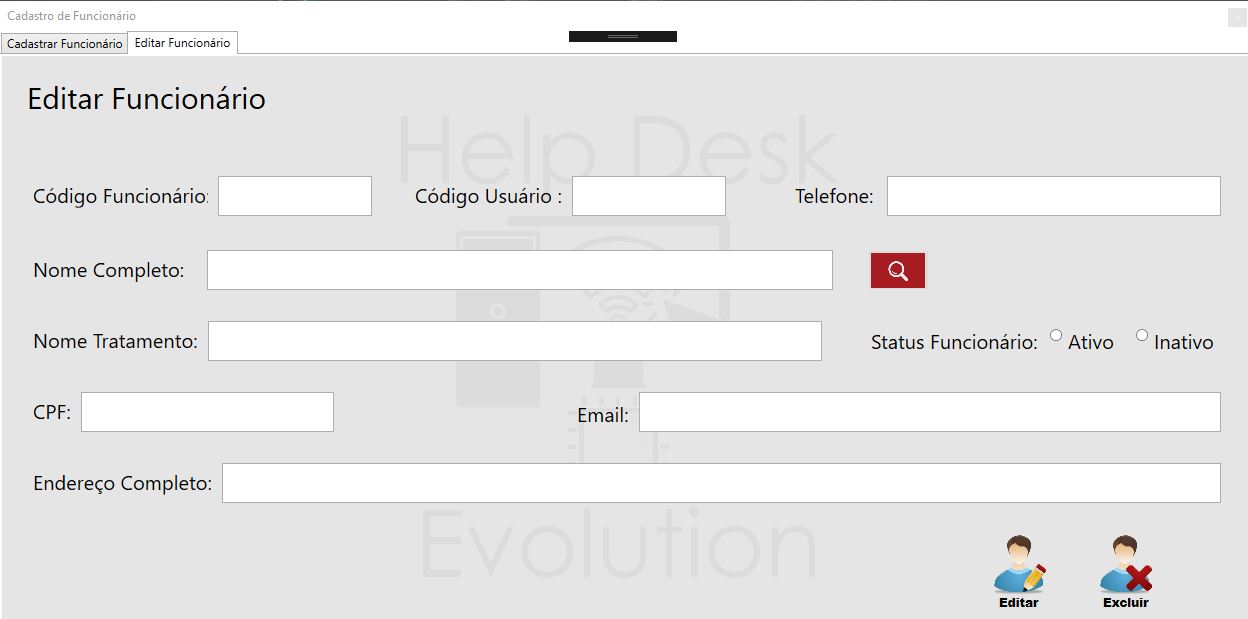
Fonte:M.I.J.H Systens

Figura 15-Tela de cadastrar funcionário



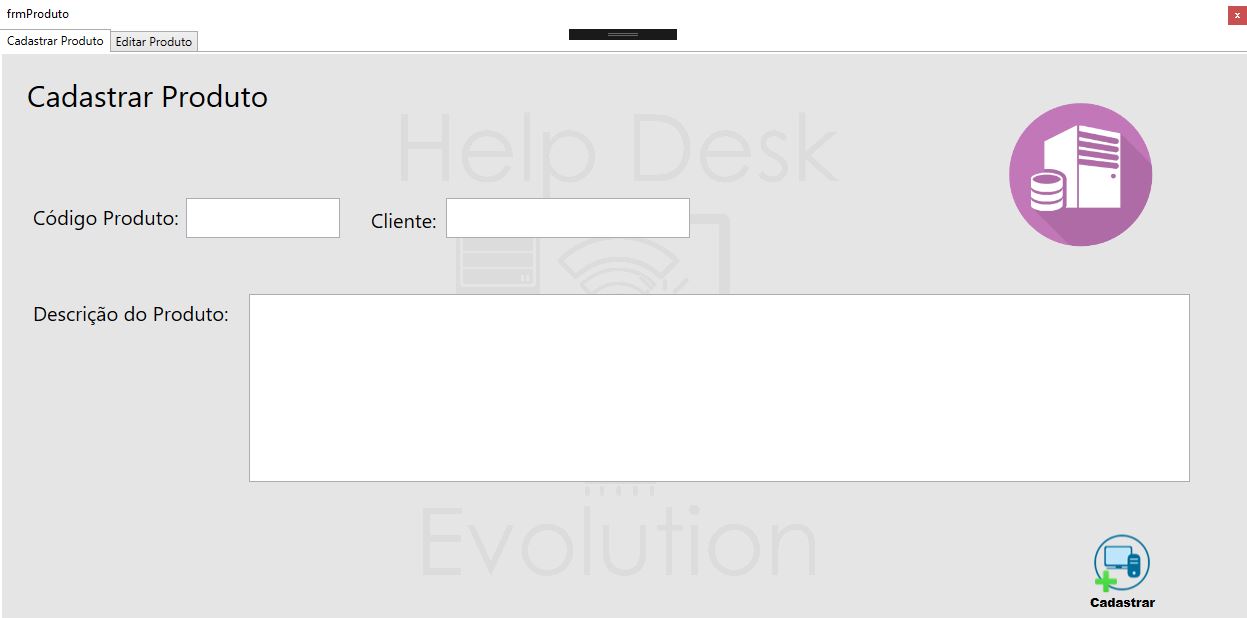
Fonte:M.I.J.H Systens

Figura 16-Tela de editar funcionário



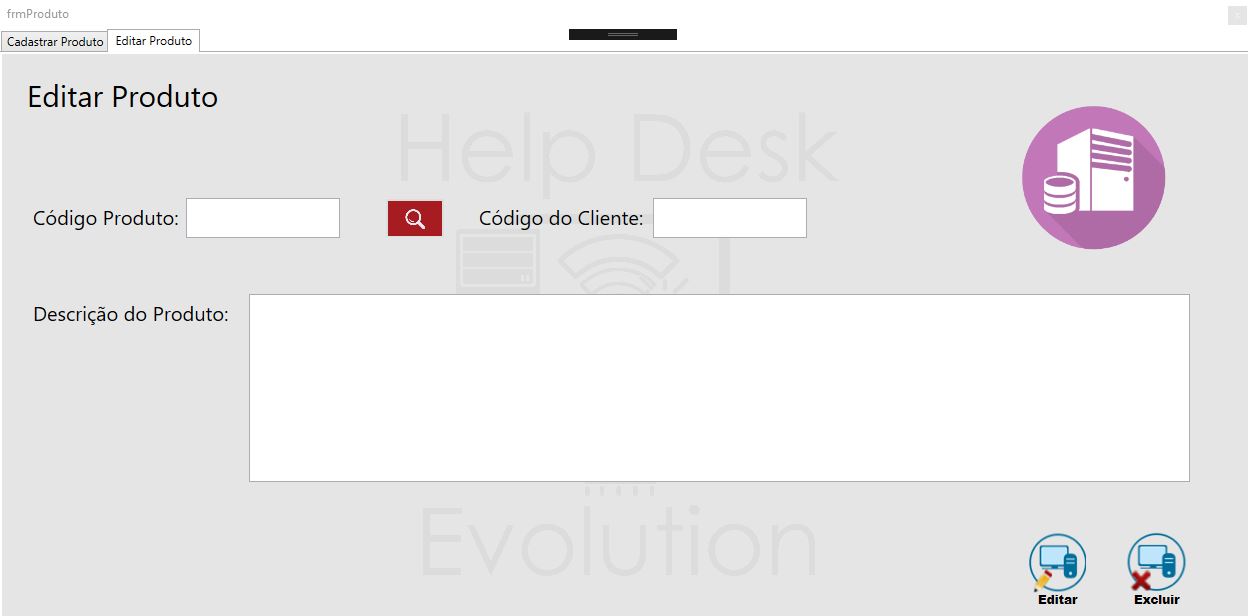
Fonte:M.I.J.H Systens

Figura 17-Tela de cadastrar produto



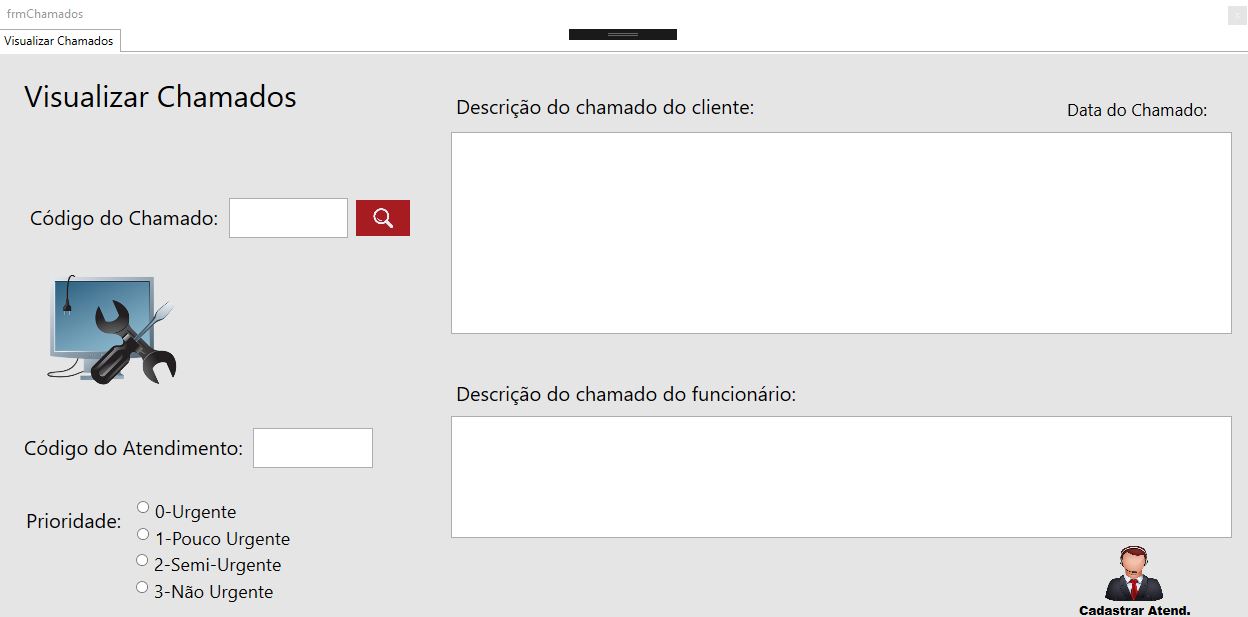
Fonte:M.I.J.H Systens

Figura 18-Tela de editar produto



Fonte:M.I.J.H Systens

Figura 19-Tela de cadastrar e visualizar atendimento



Fonte:M.I.J.H Systens

## 3.2 Web

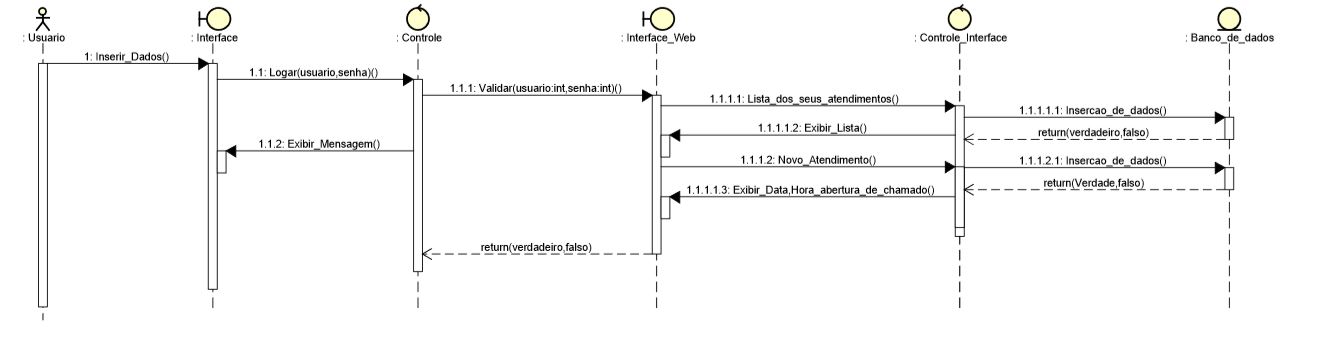
Este programa foi desenvolvido em ASP .Net, HTML, e JavaScript, toda a parte backend foi feita em ASP, utilizando o framework entity onde ele mape as classes e tabelas do banco, criando interação entre si em JavaScript foi utilizado jquery e Tooastr, para fazer a interação e comunicação das páginas, foi utilizado bootstrap para o HTML.

Foi utilizada a arquitetura MVC (model view controller), que divide a representação da informação do usuário com ele. É normalmente usado para o desenvolvimento de [interfaces de usuário](https://pt.wikipedia.org/wiki/Interface_do_usu%C3%A1rio) que divide uma aplicação em três partes interconectadas. Isto é feito para separar representações de informação internas dos modos como a informação é apresentada para e aceita pelo usuário.

Está é a parte web é aonde o cliente irá solicitar chamadas ao suporte, tudo por via web, este modulo terá a função para o cliente solicitar os serviços dos técnicos sobre um produto de sua empresa

## 3.2.1 Diagrama de sequência

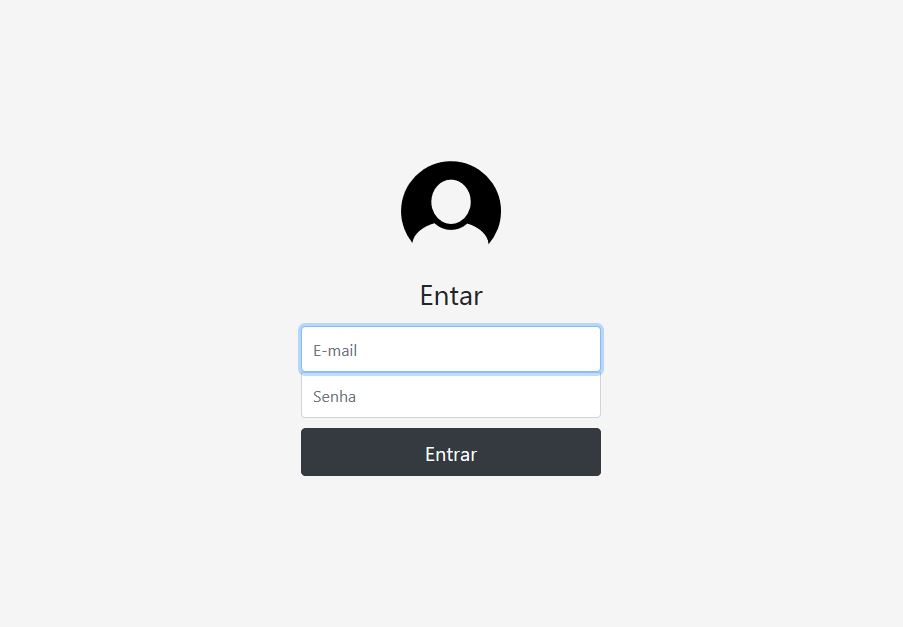
Figura 20-Diagrama de sequência do modulo web



Fonte:M.I.J.H Systens

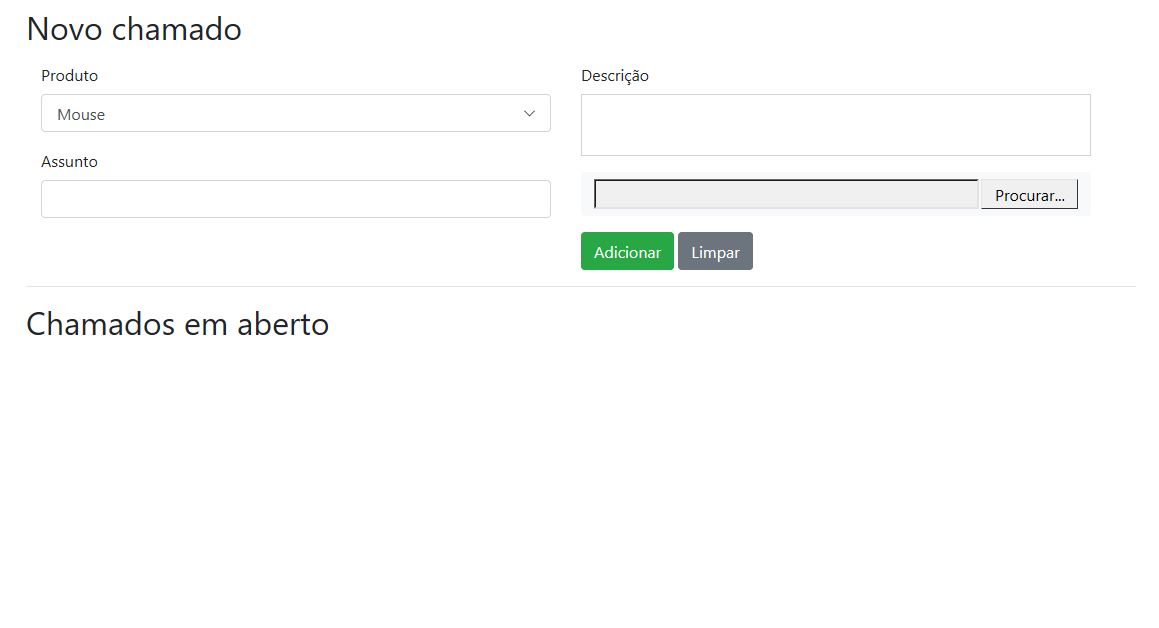
## 3.2.2 Layouts de telas

Figura 21-Tela do login do modulo web



Fonte:M.I.J.H Systens

Figura 22-Tela do cliente na web



Fonte:M.I.J.H Systens

Figura 23-Tela do funcionário na web



Fonte:M.I.J.H Systens

# 4.0 Desenvolvimento do projeto

Este projeto foi desenvolvido usando o programa Visual Studio, utilizando C# para a parte desktop e ASP.NET para a parte web.

## 4.1 Código da parte desktop

Os programas fontes desenvolvidos na parte desktop está presente no anexo 1 e no github (https://github.com/MurioLima/PIM-2018-2)

## 4.2 Código da parte web

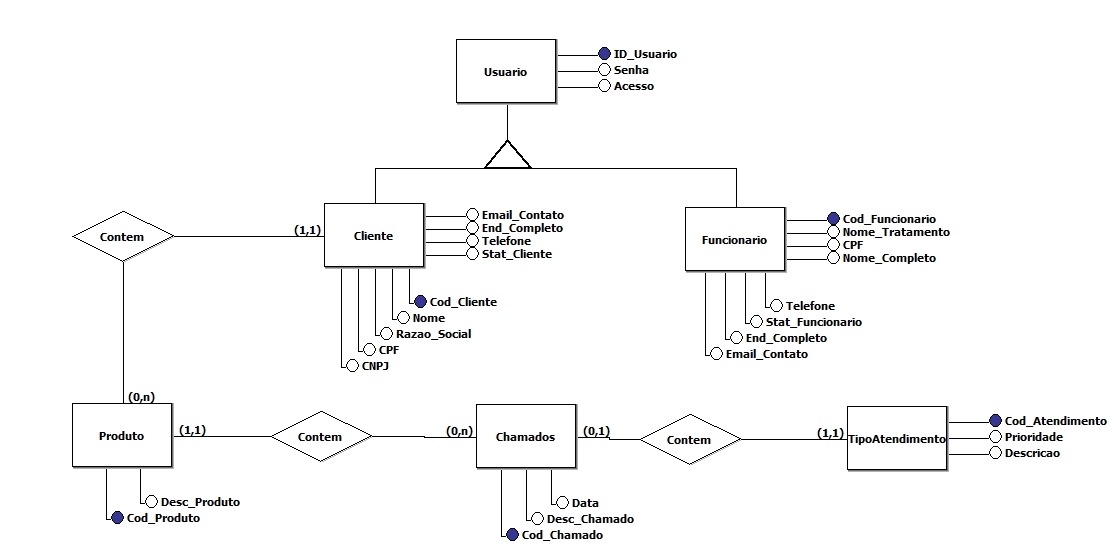
Os programas fontes desenvolvidos na parte web está presente no anexo 2 e no github (https://github.com/MurioLima/PIM-2018-2)

# 5.0 Banco de dados

Este sistema foi desenvolvido usando um banco de dados MySQL Server, e suas alterações e manutenção é feita através das ferramentas de SGBD do programa SQL Server Management Studio

## 5.1 Modelo Conceitual

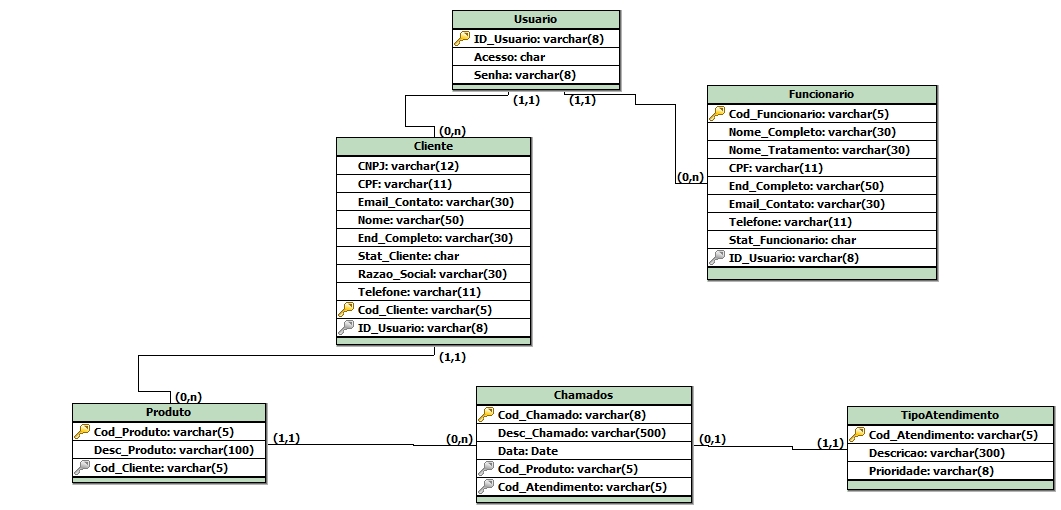
Figura 24-Modelo conceitual



Fonte:M.I.J.H Systens

## 5.2 Modelo Logico

Figura 25-Modelo logico



Fonte:M.I.J.H Systens

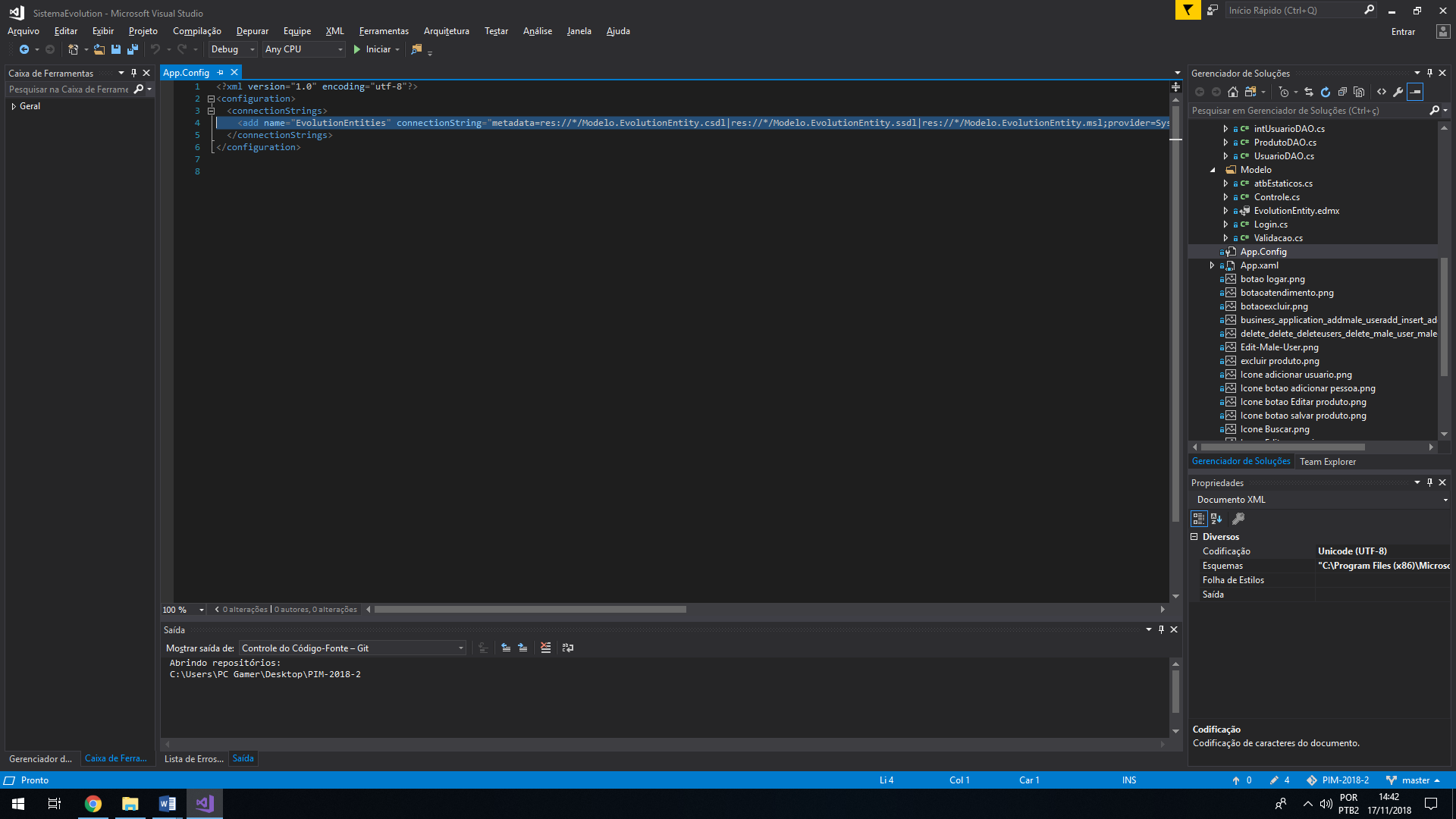
## 5.3 Modelo Físico

O Modelo Físico, ou seja, o Script do banco de dados usados no projeto Evolution está presente no anexo 3.

## 5.4 Conexões com o banco de dados

## 5.4.1 Conexão desktop

Figura 26-APP.Config



Fonte:M.I.J.H Systens

No arquivo App.Config há connectionString basta editar com as informações do seu banco:

Data Source = Endereço do banco

Catalog = Nome do banco de dados

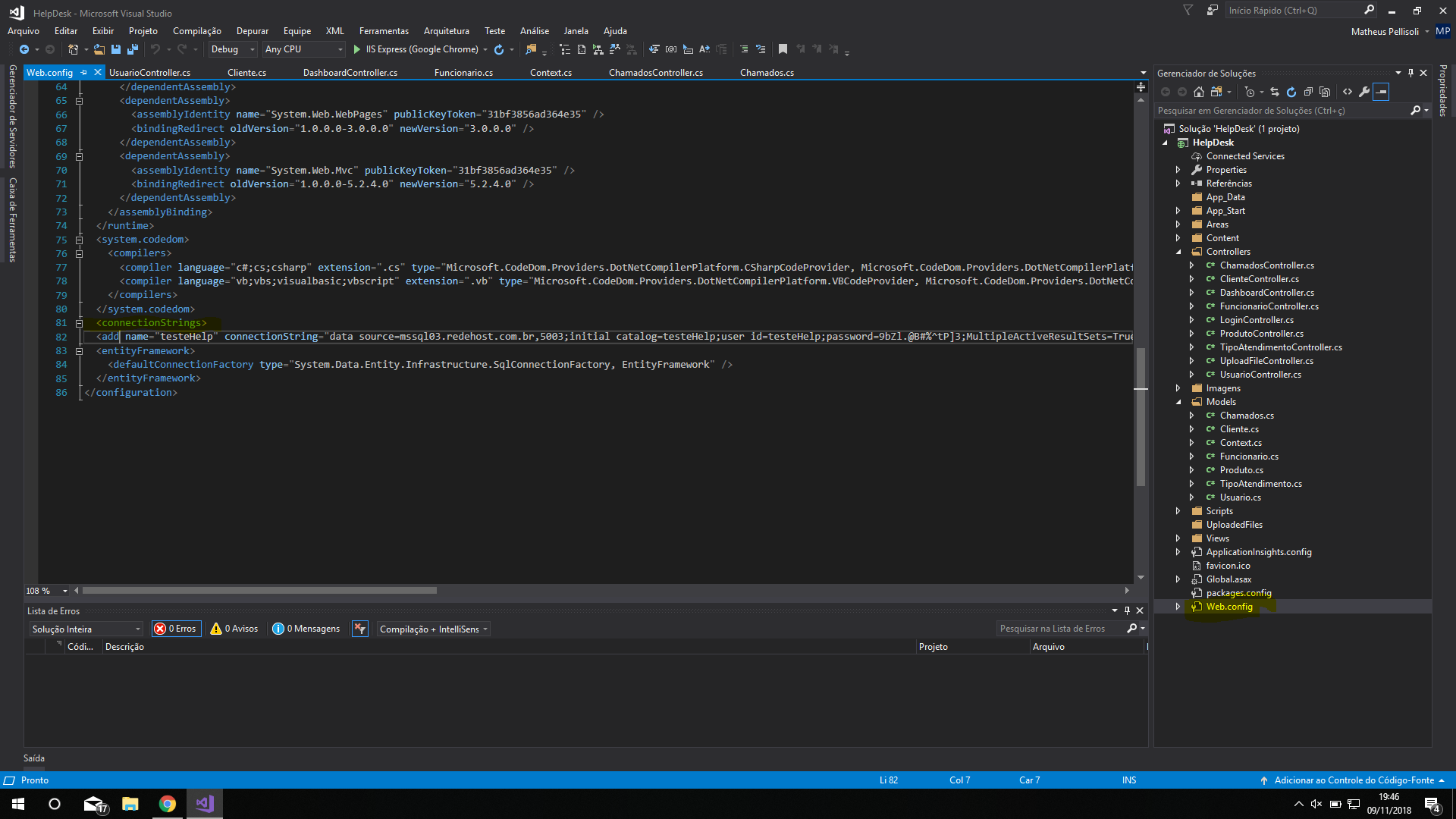
Id = Nome do usuário

Password = Senha do banco

Login utilizando autenticação do Windows = Integrated Security=True.

## 5.4.2 Conexão Web

Figura 27-WEB.Config



Fonte:M.I.J.H Systens

No arquivo Web.config há connectionString basta editar com as informações do seu banco

source = Endereço do banco

catalog = Banco de dados

id = nome do usuário

password = senha do banco

# 6.0 Requisitos de uso e desenvolvimento

Requisitos mínimos para o cliente e funcionário acessar o site web:

* Windows 7, Windows 8, Windows 8.1, Windows 10 ou versões posterior
* Processador Intel Pentium 4 ou posterior compatível com SSE2
* Processador de 32 bits (x86) ou 64 bits (x64) de 1 GHz ou superior\*
* 1 GB de RAM (32 bits) ou 2 GB de RAM (64 bits)
* 16 GB (32 bits) ou 20 GB (64 bits) de espaço em disco disponível

Requisitos mínimos para o desenvolvimento do sistema:

* Windows 10 versão 1507 ou superior: Home, Professional, Education e Enterprise (LTSC e S não são suportados)
* Windows Server 2016: Padrão e Datacenter
* Windows 8.1 (com [Atualização 2919355](https://support.microsoft.com/kb/2919355) ): Core, Professional e Enterprise
* Windows Server 2012 R2 (com a [atualização 2919355](https://support.microsoft.com/kb/2919355) ): Essentials, Standard, Datacenter
* Windows 7 SP1 (com as últimas atualizações do Windows): Home Premium, Professional, Enterprise, Ultimate.

Hardware:

* Processador de 1,8 GHz ou mais rápido. Dual-core ou melhor recomendado
* 2 GB de RAM; recomenda-se 4 GB de RAM (mínimo de 2,5 GB se for executado em uma máquina virtual)
* Espaço no disco rígido: até 130 GB de espaço disponível, dependendo dos recursos instalados; instalações típicas requerem 20 a 50 GB de espaço livre.
* Velocidade do disco rígido: para melhorar o desempenho, instale o Windows e o Visual Studio em uma unidade de estado sólido (SSD).
* Placa de vídeo que suporta uma resolução mínima de exibição de 720p (1280 por 720); O Visual Studio funcionará melhor em uma resolução de WXGA (1366 por 768) ou superior.

Banco de dados

* Requisitos para uso ou desenvolvimento
* Memória recomendado: Edições Express: 1 GB
* Qualquer outro tipo de memória deve conter no mínimo 4 GB e deve ter o aumento gradativo conforme o banco de dados cresce.
* Velocidade do processador recomendável: 2,0 GHz ou mais rápido
* Tipo de processador: processador x64, AMD Opteron, AMD Athlon 64, Intel Xeon com suporte Intel EM64T, Intel Pentium IV com suporte EM64T

# 7.0 Parametros de Qualidade

## 7.1 Funcionalidades

## 7.1.2 Adequação

O sistema Help-desk Evolution é possui dois módulos, um desktop para os funcionários, e um modulo web, feito para o uso dos clientes. O modulo desktop é onde acontece a administração de contas de clientes e funcionários, produtos e chamados, já o modulo web é feito somente para que o cliente solicite chamadas, portanto cada um dos módulos possui somente funcionalidades necessárias.

## 7.1.3 Acurácia

O sistema Help-desk Evolution é feito para prestar serviços de suporte técnico, os dados utilizados no sistema provêm de um banco de dados, ele exibe tais dados perfeitamente.

## 7.1.4 Interoperabilidade

O sistema Help-desk Evolution é um sistema único, com seu próprio banco de dados, que não possui interação com outros sistemas ou banco de dados.

## 7.1.5 Segurança

O sistema Help-desk Evolution, possui logins tanto para o módulo desktop quanto para o módulo web, o login do desktop possui uma checagem de dados de login e senha, para pessoas que não sejam funcionários possam logar , e o login da web também possui uma checagem de login e senha, para que clientes que não tenham um registro possam logar.

## 7.2 Confiabilidade

## 7.2.1 Maturidade

O sistema Help-desk Evolution é projetado para que não haja falhas decorrentes ao seu uso, e contém uma série de validações para que não ocorra erros relacionados os dados de seu banco de dados.

## 7.2.2 Tolerância a falhas

Não se aplica ao projeto.

## 7.2.3 Recuperabilidade

Caso haja algum tipo de falha com o sistema ele terá que ser reinstalado, para que não haja perda de dados, é recomendado que se tenha um backup do sistema e do banco de dados que esteja sendo utilizado.

## 7.3 Usabilidade

## 7.3.1 Inteligibilidade

O sistema Help-desk Evolution, em seus dois módulos foi desenvolvido com o objetivo de ser claro e intuitivo, tanto para os funcionários que trabalharam com desktops quanto para os clientes que acessarão a parte web, os seu layout é claro e fácil de compreender,

## 7.3.2 Apreensibilidade

O sistema Help-desk Evolution é de fácil aprendizado tanto para os funcionários que trabalharão com ele quanto para os clientes que o utilizaram.

## 7.4 Manutenibilidade

## 7.4.1 Analisabilidade

Os códigos do sistema Help-desk Evololution foram desenvolvidos com um padrão de orientação a objetos, para o modulo desktop foi desenvolvido com uma arquitetura de camadas, e o modulo web foi utilizado a arquitetura MVC, para que haja facilidade em análises futuras.

## 7.4.2 Modificabilidade

O sistema Help-desk Evolution foi desenvolvido em um padrão de orientação a objetos, para o modulo desktop foi desenvolvido com uma arquitetura de camadas, e o modulo web foi utilizado a arquitetura MVC, para facilitar mudanças futuras.

## 7.4.3 Estabilidade

O sistema Help-desk Evolution, foi desenvolvido para que opere com estabilidade, porem sua parte web só pode ser acessada pelo cliente caso haja uma conexão estável com a internet.

# 8.0 Testes

## 8.1 Testes do modulo desktop

**TESTE LOGIN SEM SUCESSO (CASO-1):**

Login sem sucesso por Erro da Senha.

**Objetivo do Teste:**

Testar o login de um usuário existente informando senha inválida para ele.

**Técnica utilizada:**

Digitação e análise dos resultados através da observação do testador.

**Estratégias:**

Utilizar o login do produto principal desenvolvido para ambiente desktop.

Utilizar a base de dados de testes.

**Ferramentas Necessárias:**

Computador PC e sistema operacional Windows 8.1 Pro ou 7 Pro.

Word para acesso a documentação.

**Condições de Teste:**

O usuário “igor@gmail.com” está cadastrado no banco com a senha “12345”

**Critério de Êxito**

Não permitir acessar o sistema com uma senha inválida para o usuário.

**Resultado:**

Satisfatório.

**Observações:** Embora o foco não tenha ficado como o previsto o objetivo do teste foi atingido, impedindo o login e limpando os campos para outra tentativa de login.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Procedimento de Teste | # | Resultado Esperado | Resultado Obtido |
| 1 | Na tela de login do sistema digitar:  Usuário: “Igor@gmail.com” Senha: “57458” | 2 | O sistema exibe uma mensagem: “Login ou Senha incorretos” | Mensagem exibida conforme o esperado |
| 3 | Clicar no botão [OK] da mensagem | 4 | O sistema exibe novamente a tela de login | Tela de login exibida conforme o esperado |

**Identificação do Testador:** Igor Mendes

**TESTE CADASTRAR CLIENTE SEM SUCESSO (CASO-2):**

Cadastrar cliente sem sucesso por inserir um código cliente existente.

**Objetivo do Teste:**

Testar o cadastramento de um cliente informando um código cliente já existente no banco de dados.

**Técnica utilizada:**

Digitação e análise dos resultados através da observação do testador.

**Estratégias:**

Utilizar a tela de cadastro do cliente desenvolvido para ambiente desktop.

Utilizar a base de dados de testes.

**Ferramentas Necessárias:**

Computador PC e sistema operacional Windows 8.1 Pro ou 7 Pro.

Word para acesso a documentação.

**Condições de Teste:**

O cliente “Mendes” está cadastrado no banco com o código cliente “2686”

**Critério de Êxito**

Não permitir cadastrar o novo cliente utilizando o mesmo código cliente.

**Resultado:**

Satisfatório.

**Observações:** O objetivo do teste foi atingido

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Procedimento de Teste | # | Resultado Esperado | Resultado Obtido |
| 1 | Na tela de cadastro do sistema digitar:  Código cliente: “2686”  Nome Fantasia: “Systems”  E-mail: “systems@gmail.com” | 2 | O sistema exibe uma mensagem: “Código do cliente já cadastrado, digite outro código.” | Mensagem exibida conforme o esperado |
| 3 | Clicar no botão [OK] da mensagem | 4 | O sistema exibe novamente a tela de cadastro para corrigir o código cliente. | Tela de cadastro exibida para corrigir o código cliente conforme o esperado. |

**Identificação do Testador**: Igor Mendes

**TESTE CADASTRAR FUNCIONÁRIO SEM SUCESSO (CASO-3):**

Cadastrar Funcionário sem sucesso por inserir um código funcionário existente.

**Objetivo do Teste:**

Testar o cadastramento de um funcionário informando um código funcionário já existente no banco de dados.

**Técnica utilizada:**

Digitação e análise dos resultados através da observação do testador.

**Estratégias:**

Utilizar a tela de cadastro do cliente desenvolvido para ambiente desktop.

Utilizar a base de dados de testes.

**Ferramentas Necessárias:**

Computador PC e sistema operacional Windows 8.1 Pro ou 7 Pro.

Word para acesso a documentação.

**Condições de Teste:**

O funcionário “Murilo” já está cadastrado no banco com o código cliente “888”.

**Critério de Êxito**

Não permitir cadastrar o novo funcionário utilizando o mesmo código funcionário.

**Resultado:**

Satisfatório.

**Observações:** O objetivo do teste foi atingido

**Identificação do Testador**: Igor Mendes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Procedimento de Teste | # | Resultado Esperado | Resultado Obtido |
| 1 | Na tela de cadastro do sistema digitar:  Código funcionário: “888”  Nome Completo: “Murilo”  E-mail: “murilo@gmail.com” | 2 | O sistema exibe uma mensagem: “Código do funcionário já cadastrado, digite outro código.” | Mensagem exibida conforme o esperado |
| 3 | Clicar no botão [OK] da mensagem | 4 | O sistema exibe novamente a tela de cadastro para corrigir o código funcionário. | Tela de cadastro exibida para corrigir o código funcionário conforme o esperado. |

**TESTE CADASTRAR PRODUTO SEM SUCESSO (CASO-4):**

Cadastrar Produto sem sucesso por inserir um código cliente não existente ou deixar vazio.

**Objetivo do Teste:**

Testar o cadastramento de um produto informando um código cliente não existente no banco de dados ou deixar vazio.

**Técnica utilizada:**

Digitação e análise dos resultados através da observação do testador.

**Estratégias:**

Utilizar a tela de cadastro do cliente desenvolvido para ambiente desktop.

Utilizar a base de dados de testes.

**Ferramentas Necessárias:**

Computador PC e sistema operacional Windows 8.1 Pro ou 7 Pro.

Word para acesso a documentação.

**Condições de Teste:**

O funcionário “Murilo” está cadastrado no banco com o código cliente “888”

**Critério de Êxito**

Não permitir cadastrar o novo produto por não informar um código cliente valido ou deixar vazio.

**Resultado:**

Satisfatório.

**Observações:** O objetivo do teste foi atingido

**Identificação do Testador**: Igor Mendes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Procedimento de Teste | # | Resultado Esperado | Resultado Obtido |
| 1 | Na tela de cadastro do sistema digitar:  Código Produto: “001”  Código Cliente: “0102”  Descrição Produto: “Mouse” | 2 | O sistema exibe uma mensagem: "Código do cliente já cadastrado ou vazio." | Mensagem exibida conforme o esperado |
| 3 | Clicar no botão [OK] da mensagem | 4 | O sistema exibe novamente a tela de cadastro para corrigir o código cliente. | Tela de cadastro exibida para corrigir o código cliente conforme o esperado. |

**CADASTRAR ATENDIMENTO SEM SUCESSO (CASO-5):**

Cadastrar atendimento sem sucesso por inserir um código de atendimento existente.

**Objetivo do Teste:**

Testar o cadastramento de um atendimento informando um código do atendimento existente no banco de dados.

**Técnica utilizada:**

Digitação e análise dos resultados através da observação do testador.

**Estratégias:**

Utilizar a tela de cadastro do cliente desenvolvido para ambiente desktop.

Utilizar a base de dados de testes.

**Ferramentas Necessárias:**

Computador PC e sistema operacional Windows 8.1 Pro ou 7 Pro.

Word para acesso a documentação.

**Condições de Teste:**

Um atendimento já está cadastrado no banco de dados com código do atendimento ”022”.

**Critério de Êxito**

Não permitir cadastrar um novo atendimento por informar um código do atendimento já existente no banco de dados.

**Resultado:**

Satisfatório.

**Observações:** O objetivo do teste foi atingido

**Identificação do Testador**: Igor Mendes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Procedimento de Teste | # | Resultado Esperado | Resultado Obtido |
| 1 | Na tela de chamados do sistema digitar:  Código do atendimento: “022”  Prioridade: “1”  Descrição: “Defeito no cabo” | 2 | O sistema exibe uma mensagem: "Código do Atendimento já cadastrado, digite outro código." | Mensagem exibida conforme o esperado |
| 3 | Clicar no botão [OK] da mensagem | 4 | O sistema exibe novamente a tela de chamados para corrigir o código atendimento. | Tela de chamados exibida para corrigir o código atendimento conforme o esperado. |

|  |  |  |  |
| --- | --- | --- | --- |
| Identificação ou codificação: | | Nome do teste: Desktop teste | |
| Área a ser testada: Sistema desktop | | | |
| Técnica: Teste V&V | Data: 15/ 11/ 2018 | | Versão:1.0 |
| Objetivo: Testar o sistema desktop segundo as técnicas V&V. | | | |
| Ferramentas necessárias: um computador, Word para acesso a documentação. | | | |
| * **Teste de unidade:** Foi aplicativo um total de 2 horas de testes procurando por erros de lógica e implementação em cada módulo, separadamente, validando dados válidos e inválidos via entrada/saída, os testes teve resultados positivos, toda a lógica do sistema se comportou como o projetado. * **Teste de integração:** Foi aplicado um total de 3 horas de testes com foco na integração entre as unidades do sistema, procurando falhas de envio e recebimento de dados, no tempo total do teste, não foi apresentado nenhuma falha neste requisito, todos os dados enviados ou recebidos do banco de dados se mostrou 100% eficaz. * **Teste de sistema:** O sistema foi testado no total de 6 horas de uso, todo o sistema funcionou 100% conforme o esperado, as telas se interagem corretamente, componentes são compatíveis, transferência de dados do sistema para o banco de dados se comportou-se muito bem, o resultado da avaliação foi totalmente positiva, dentro dos parâmetros esperados. * **Teste de segurança:** Foi feito uma serie de tentativas de login com senha ou usuário incorretos, o programa negou o acesso com êxito, pedindo o usuário verificar se está correta as informações. * **Teste de estresse**: Foi feito um teste de 5 funcionários acessando a parte desktop ao mesmo tempo, não ocorreu problemas, o teste foi básico pois raramente vários funcionários irão acessar o sistema ao mesmo tempo. * **Teste de desempenho:** No teste de desempenho foi feito o teste de velocidade em todas as telas e que se mostraram em uma média geral de para o envio e o recebimento dos dados que está dentro da margem de aceitação dos usuários, a velocidade em média é de 1s para o carregamento das telas e 2s para enviar e receber do banco de dados. * **Teste de caminho:** não se aplica. * **Teste de regressão:** não se aplica. * **Teste de caixa branca:** Foi aplicado umas séries de testes rigorosos em partes especificas examinando caminhos lógicos essenciais do sistema, verificando os componentes, exercitando os conjuntos específicos de condições e/ou ciclos do sistema, o código se mostrou totalmente executável nas considerações do analista. * **Teste de caixa preta:** não se aplica. | | | |
|  | | | |

# 9.0 Matérias do Semestre

## 9.1 Programação Orientada a Objetos 2

A matéria de POO 2 continua a mostrar as técnicas da programação orientada a objetos, começando em como realmente trabalhar em um projeto usando apenas objetos, também ensinar em como trabalhar com banco de dados nos projetos , mostra como executar a conectividade com os bancos , ensina a linguagem sql,ensina como trabalhar com entity framework e as instruções em linq e gerenciamento de relatórios.

## 9.2 Tópicos especiais de programação orientada a objetos

A meteria de top esp de programação orientada a objetos foca no desenvolvimento de projetos em Java, e é introduzido o software android studio para a produção de aplicativos para dispositivos android, são ensinados as diferença entre as instruções de Java e C#, as ferramentas do android studio, funções de banco de dados do android studio e como desenvolver aplicativos com seus bancos de dados.

## 9.3 Desenvolvimento de Software para Internet

A matéria de desenvolvimento de software para internet, ensina o desenvolvimento de softwares para web, ensina as linguagens usadas para a programação web, mostra os conceitos e características dos softwares produzidos para web, ensina o conceitos da arquitetura MVC , e o desenvolvimento usando esta arquitetura, mostra os conceitos de API e como desenvolver usando APIS, mostra como utilizar pacotes como entity framework e web client e de como implementa-los no desenvolvimento dos projetos.

## 9.4 Projeto Orientado a Objetos

A meteria de Projeto Orientado a Objetos, ensina os conceitos do que é um projeto orientado a objetos, ensina os conhecimentos básicos sobre as metodologias e técnicas de projetos utilizados para o desenvolvimento de projetos, introduz as tecnologias de apoio para a produção de projetos.

## 9.5 Gerenciamento de projeto de software

A matéria de Gerenciamento de projeto de software, ensina os conceitos e técnicas de controle de projeto, ciclo de vida dos projetos, escopo, prazos e custos, gerenciamento de riscos e gestão de projetos de desenvolvimento de software.

# 10 Conclusão

De acordo com todos os processos de produção que foram utilizados para a conclusão do sistema Evolution, foi possível adquirir o conhecimento de como realmente são os processos de produção de um sistema na área de trabalho. Ao decorrer do desenvolvimento do sistema, foi possível ver como os componentes do curso se encaixam no meio de produção de um sistema.

No desenvolvimento do sistema Evolution, foi possível ver a utilização dos conteúdos de todas as matérias que foram dadas ao longo do curso em prática, começando com tudo que é relacionado ao gerenciamento de projetos e documentação, de como gerenciar um projeto com melhor eficiência, trabalhar com vários tipos de diagramas, fluxogramas, UML e seus artefatos, utilizando logica e técnicas de programação, aplicação de arquiteturas para melhor eficiência do sistema, gerenciamento de banco de dados, trabalhar com módulos em diferentes linguagens, para enfim desenvolver o sistema Help-Desk Evolution.

Com as pesquisas realizadas que complementaram nosso aprendizado no semestre, podemos observar que são várias atividades na execução de um projeto de sistema, tivemos bom conhecimento no assunto abordado no projeto e conseguimos produzir o produto final dentro do tempo estipulado pelo cliente, e com tudo isso podemos ver em pratica todo o conteúdo do curso de analise e desenvolvimento de sistema.

# Referencias

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR ISO/IEC **9126-1 Engenharia de software - Qualidade de produto - Parte 1: Modelo de qualidade**. 2003

<https://support.microsoft.com/pt-br/help/10737/windows-7-system-requirements>

<https://support.google.com/chrome/a/answer/7100626?hl=pt-BR>

<https://docs.microsoft.com/en-us/visualstudio/productinfo/vs2017-system-requirements-vs>