

**CENTRO PAULA SOUZA  
ESCOLA TÉCNICA ESTADUAL IRMÃ AGOSTINA  
Curso Técnico em Informática**

**Barbara Rodrigues da Silva  
Gustavo da Cunha Avila  
João Victor Silva Cepelo  
Lucas Chagas da Silva  
Luiz Felipe de Paiva**

**MUSITEC**

**São Paulo  
2016**

**Barbara Rodrigues da Silva  
Gustavo da Cunha Avila  
João Victor Silva Cepelo  
Lucas Chagas da Silva  
Luiz Felipe de Paiva**

## **MUSITEC**

Projeto a ser apresentado na disciplina de Planejamento de Trabalho de Conclusão de Curso como requisito básico para a apresentação do Trabalho de Conclusão de Curso do Técnico em Informática.

Orientador: Edídio Lima

**São Paulo - SP  
2016**

**Barbara Rodrigues da Silva  
Gustavo da Cunha Avila  
João Victor Silva Cepelo  
Lucas Chagas da Silva  
Luiz Felipe de Paiva**

## **MUSITEC**

Trabalho de Conclusão de Curso  
submetido à Etec Irmã Agostina, como parte  
dos requisitos necessários para a obtenção  
do certificado de técnico em informática.

**São Paulo - SP  
2016**

*Dedicamos este trabalho à  
todas a pessoas que gostariam  
de aprender teoria musical.*

## AGRADECIMENTOS

*Agradecemos aos integrantes do grupo pela participação e esforço para que fosse possível a realização e finalização deste projeto.*

*Agradecemos também às pessoas que acreditaram em nós e nos incentivaram a continuar, assim como os professores que nos orientaram e nos guiaram até esse caminho.*

## LISTA DE FIGURAS

FIGURA 1 - PESQUISA DE FAIXA ETÁRIA.....	14
FIGURA 2 - PESQUISA - GRÁFICO DE SEXO DO PÚBLICO-ALVO E APROVAÇÃO.....	14
FIGURA 3 - PESQUISA - GRÁFICO DE INTERESSE EM INSTRUMENTOS.....	15
FIGURA 4 - PESQUISA - GRÁFICO DE ESTUDO .....	16
FIGURA 5 – FLUXOGRAMA.....	17
FIGURA 6 - DIAGRAMA ENTIDADE RELACIONAMENTO DO BANCO DE DADOS .....	32
FIGURA 7 - CASO DE USO .....	33

## LISTA DE TABELAS

TABELA 1 - CADASTRO .....	34
TABELA 2 - LOGIN .....	34
TABELA 3 - PERFÍL .....	34
TABELA 4 - INSTRUMENTO.....	35
TABELA 5 - TEORIA.....	35
TABELA 6 - TESTE FINAL .....	36

# SUMÁRIO

<b>INTRODUÇÃO</b>	9
<b>ABSTRACT</b>	11
<b>2 MODELAGEM CONCEITUAL</b>	13
2.1 Pesquisa de Campo	13
2.2 Metodologia	16
2.3 Fluxograma	17
<b>3 FUNCIONAMENTO DO SISTEMA</b>	18
3.1 Descrição Método por método	18
3.1.1 Classe Cadastro.cs	18
3.1.2 Classe Conexao.cs	19
3.1.3 Classe dal.cs	19
3.1.4 Classe frmAux1.cs	20
3.1.5 Classe frmCadastro.cs	21
3.1.6 Classe frmLogin.cs	23
3.1.7 Classe metodo.cs	25
3.1.8 Classe modulo1.cs	27
3.2 Tecnologias empregadas	30
3.2.1 Criação do nome	31
<b>4 MODELAGEM LÓGICA</b>	32
4.1 Diagrama Entidade Relacionamento	32
4.1.1 Relacionamentos	33
4.1.2 Diagrama de Caso de Uso	33
4.2 Especificação de Caso de Uso	33
<b>5 NOTAS DE VERSÃO</b>	37
<b>CONCLUSÃO</b>	39
<b>REFERÊNCIAS</b>	40



## INTRODUÇÃO

A tecnologia hoje em dia vem evoluindo de uma forma bastante significativa, e cada vez mais pessoa têm acesos a Internet, principalmente com celulares. Com base nisso, podemos observar que o mercado de jogos foi se expandindo de uma forma drástica, e com isso os jogos educacionais também.

A partir desse ponto de vista, tivemos a ideia de fazer um projeto que ensina a teoria musical de uma forma intuitiva, pois muitos jogos considerados educacionais são feitos apenas com o intuito de aprender, porém não com uma boa didática.

Com referência no livro de Maura Penna: “Música e Seus Ensinos”, escolhemos fazer nosso projeto em forma de aplicativo e que leve qualidade e educação musical para as pessoas, visando o público infanto-juvenil.

Nem todos os sistemas oferecem conteúdos com grande quantidade de temas e ensinam a teoria completa gratuitamente e de uma forma dinâmica/intuitiva. Além de muitas pessoas não terem condições financeiras para pagar um professor ou um curso de música ou um conservatório. O que acaba deixando essas pessoas sem muitas opções.

Ainda de acordo com o livro da escritora Maura Penna, professora do departamento de educação musical da Universidade Federal da Paraíba, graduada em Música, Educação Artística e Ciências Sociais, em “Música e seus ensinos”, Maura mostra que a educação musical deveria ser implementada para crianças e adolescentes como prática de desenvolvimento cultural, artístico e social:

“Consideramos familiar aquele tipo de música que faz parte de nossa vivência; justamente porque o fazer que parte de nossa vivência permite que nós nos familiarizemos com os seus princípios de organização sonora, o que torna uma música significativa para nós. [...] a compreensão da música, ou mesmo a sensibilidade a ela, tem por base um padrão culturalmente compartilhado para a organização dos sons numa linguagem artística, padrão este que, socialmente construído, é socialmente apreendido – pela vivência, pelo contato cotidiano, pela familiarização – embora também possa ser aprendido na escola.”.

Queremos que nossos usuários entendam como a teoria musical funciona para a partir dela ter mais conhecimento e facilidade em tocar instrumentos, ler partituras, harmonizar notas, tudo isso de forma dinâmica com interações, testes

para acompanhar o progresso do aprendizado, com formas que levem o usuário a ter cada vez mais incentivo em progredir no estudo de música.

Dentro do mercado de aplicativos ou programas musicais temos muitos que ensinam sem a devida teoria: Coach Guitar, Chromatik, Drum Guru, Real Drum, iLearnPiano. Essas aplicações ensinam apenas como tocar as notas em uma melodia de determinado instrumento através de uma projeção virtual de tal instrumento, mas não explicando o contexto teórico e sim visando somente na prática, o que dificulta o entendimento de pessoas que não tem nenhuma noção de teoria musical.

## **ABSTRACT**

Nowadays, the technology has been evolving in a very significant way, and more and more people have access to Internet, especially with mobile phones. Based on that, it is possible to see that the market game has been expanded in a drastic way, so as the educational games.

From this point of view, we had an idea to make a project that teaches musical theory in an intuitive way, because a lot of games considered educational are made for learning, but without a good teaching.

Based on Maura Penna's book: "Música e Seus Ensinos", we chose to make our project as an application that brings quality and musical education to all people, aimed to children and youth.

Not all systems offer the proper content with a big quantity of themes in a freeway plus in a dynamic/intuitive way. Besides that, there are a lot of people who can't afford a teacher nor a course.

Also, according to Maura Penna's book, professor of music education department at the Federal University of Paraíba, graduated in Music, Arts Education and Social Sciences, in "Music and its teachings". Maura shows that musical education should be implemented to children and teens as a practice of cultural, artistic and social development:

"We consider family the kind of music that is part of our experience; precisely because to do that part of our experience allows us to become familiar with the principles of sound organization, which makes music significant for us. [...] The music understanding, or even sensitivity to it, is based on a culturally shared standard for the organization of sounds in language arts, a pattern that socially constructed, is socially perceived – by experience, by daily contact, by familiarization –, although it can also be learned in school."

We want our users to understand how musical theory works, and, from that, have more knowledge and easiness when playing instruments, reading sheets, harmonizing notes, all dynamically with interactions; testing to look the learning progress, with methods that take the user to have more and more incentive to progress in the study of music.

On the market, there are a lot of musical programs that teach without proper theory, such as: Coach Guitar, Chromatik, Drum Guru, Real Drum,

iLearnPiano. These applications teach only how to play the notes in a melody of an instrument through a virtual projection of the instrument, but not explaining the theoretical context, but targeting only in practice, which complicates the understanding of people who have no notion of musical theory.

# 1 MODELAGEM CONCEITUAL

Para a construção do projeto, foi feita uma pesquisa para obtermos melhor informação de como produzir o conteúdo adequado. Essa pesquisa foi feita pessoalmente com a ajuda dos professores do Auditório Ibirapuera e alunos do curso de música.

Tal pesquisa é necessária para sabermos a melhor forma de apresentar o conteúdo aos usuários que utilizarão o aplicativo do projeto, visto que o grande diferencial e a principal proposta aqui é trazer uma interatividade junto com a teoria; e a melhor forma de fazer isso é saber como os professores de verdade ensinam, assim adaptar a técnica que achamos mais conveniente para o nosso projeto.

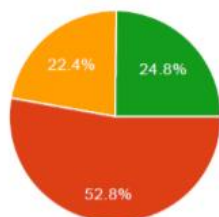
A importância dos estudantes de música é que, apesar deles já terem noção do que é teoria musical, eles podem apontar qual foram as principais dificuldades em aprender e o que eles fariam para ensinar outras pessoas.

Além da pesquisa pessoal para a produção do conteúdo, foi feita uma pesquisa através da internet para sabermos se nosso projeto é realmente necessário e vermos a aceitação dele na comunidade, principalmente para o nosso público-alvo.

## 1.1 Pesquisa de Campo

Foi feito um questionário que pôde ser respondido de forma anônima, pela internet, onde continha cinco perguntas para determinarmos melhor quem seria o público-alvo que se interessava mais no nosso projeto, o nível de interesse e quais eram os instrumentos musicais mais procurados. Tivemos a colaboração de 161 pessoas que participaram da pesquisa e com base em suas respostas obtivemos as seguintes conclusões:

Idade



Menor que 10 anos	0	0%
Entre 11 e 18 anos	85	52.8%
Entre 19 e 25 anos	36	22.4%
Maior que 26 anos	40	24.8%

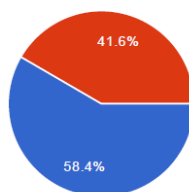
*Figura 1 - Pesquisa de faixa etária*

mostrado só nos mostra uma base de qual faixa estaria de idade o nosso projeto será aceito com mais sucesso.

A *Figura 1* é primeiro gráfico,

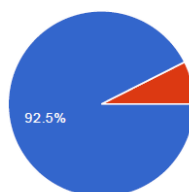
podemos notar a sua função em informar as faixas etárias de idade dos usuários, separando-as em algumas idades específicas. Com base nesse resultado podemos avaliar qual faixa etária terá mais acesso a nossa ferramenta, que no caso seria as pessoas entre 11 e 18 anos de idade. Vale lembrar que o nosso projeto é focado para crianças, jovens e adultos com qualquer idade e que tenha acesso à nossa ferramenta, o gráfico

Sexo



Feminino	94	58.4%
Masculino	67	41.6%

Você usaria um aplicativo que ensina teoria musical?

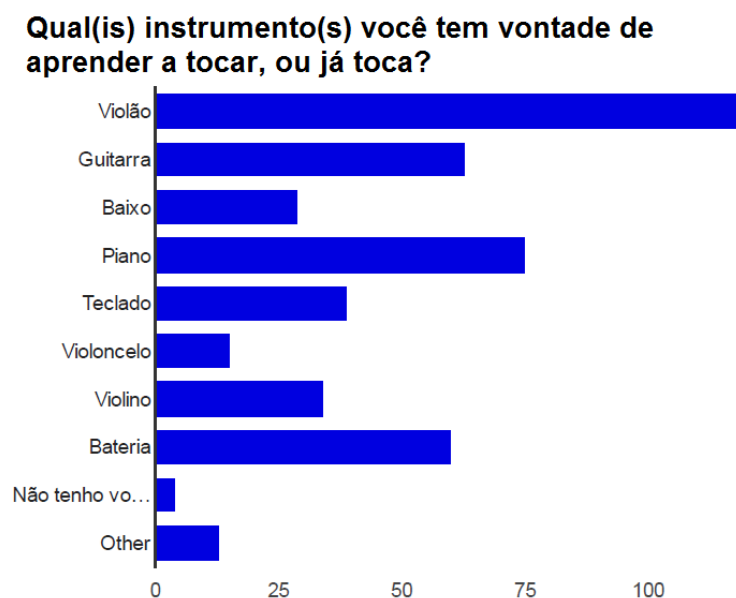


Sim	149	92.5%
Não	12	7.5%

*Figura 2 - Pesquisa - Gráfico de sexo do público-alvo e aprovação*

Vejamos agora na *Figura 2*, onde contém os dois gráficos que nos mostram dois assuntos de extrema importância para o nosso projeto, um explica qual é o sexo dos usuários que estão mais interessados em Teoria Musical, e o outro faz a pergunta principal: Você usaria uma aplicação voltada ao ensino de teoria musical? No nosso primeiro gráfico, o público feminino obteve maioria na pesquisa de 58% e se mostra mais interessado em aprender teoria musical com o nosso

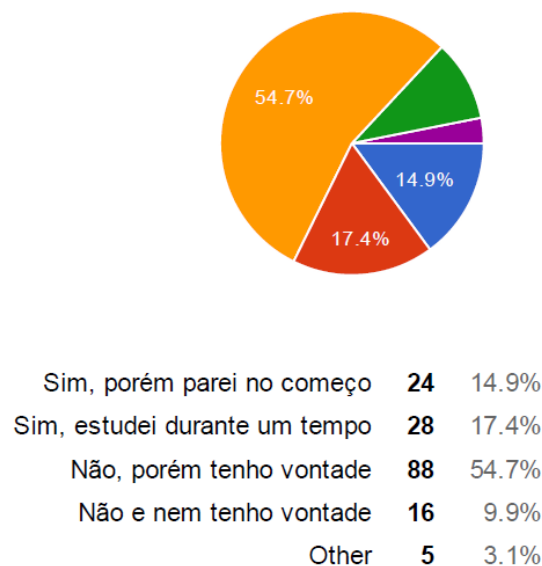
projeto. O segundo gráfico mostra a importância em ter pessoas que usem a nossa aplicação, afinal é em base dessas pesquisas que verificamos a viabilidade e aceitação do nosso projeto, sendo 92% a favor do projeto.



*Figura 3 - Pesquisa - Gráfico de interesse em instrumentos*

No gráfico acima (*Figura 3*), notamos a importância de quais instrumentos o público mais pediu na nossa pesquisa. Os instrumentos mais votados em nossas pesquisas foram: violão, piano e guitarra, e com o gráfico concluímos quais são os instrumentos mais procurados para aprendizagens, sendo os dois primeiros os instrumentos que serão inseridos no projeto.

### Você já estudou teoria musical alguma vez?



*Figura 4 - Pesquisa - Gráfico de estudo*

Com base nas 161 pessoas que responderam nosso questionário (vide *Figura 4*), notamos que 17,4% dos entrevistados já estudaram Teoria Musical em algum momento de suas vidas e 54,7% deles tem vontade de estudar sobre a teoria musical para aprender a tocar instrumentos.

## 1.2 Metodologia

Com base nas pesquisas que fizemos mais as encontradas na Internet, onde é dito que aprender através de aplicativos pode ser realmente muito eficiente, desde que usado de maneira correta, foi possível ter muitas dúvidas respondidas em relação à viabilidade do projeto.

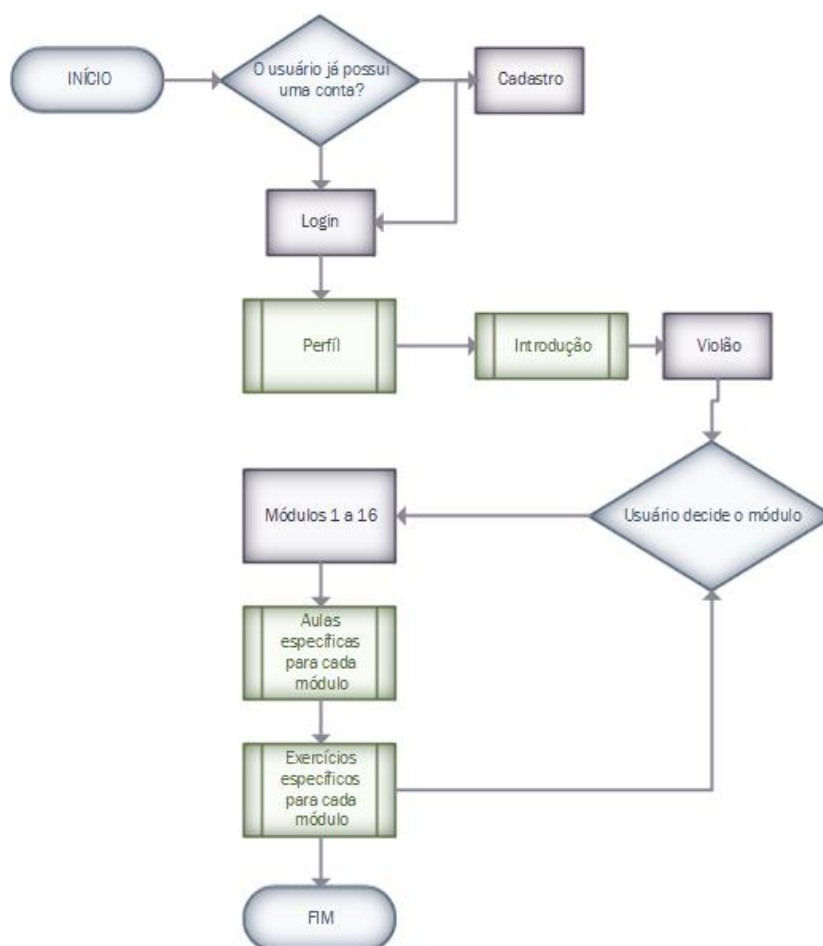
Foi evidenciado que, dependendo do estilo do projeto (no caso um aplicativo para celular) é muito eficaz a aprendizagem, pois o celular do usuário poderá ficar lembrando o dono de abrir o aplicativo e aprender jogando, também foi notado que se feito em um computador, a qualidade do projeto pode sair melhor.

Com relação ao nosso sistema, ele ajuda na aprendizagem básica da teoria musical de dois instrumentos (os que obtiveram maior demanda durante a nossa pesquisa).



### 1.3 Fluxograma

O fluxograma abaixo se refere ao funcionamento interno da aplicação, desde o momento que o usuário o acessa até concluir a suas necessidades. Na imagem podemos ver a rotina para qual a aplicação funcionara, mostrando passo a passo todas as funções disponíveis na aplicação.



*Figura 5 – Fluxograma*

É possível observar que a aplicação se preocupa com o cadastro do usuário, caso ele não possua um cadastro, ele será redirecionado à uma página específica para isso e depois voltará a tela de Login para efetua-lo, em seguida começará a com os conteúdos e depois para os testes práticos e, para encerrar e verificar se o usuário aprendeu, será feita uma “prova” para verificar sua pontuação.

## 2 FUNCIONAMENTO DO SISTEMA

### 2.1 Descrição Método por método.

#### 2.1.1 Classe Cadastro.cs

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;
```

```
namespace Musitec  
{  
    class Cadastro  
    {  
        private int nCodigo;  
        private string nNome;  
        private string nEmail;  
        private string nLogin;  
        private string nSenha;  
        private string nSexo;  
        private string nData;  
  
        public string Data  
        {  
            get { return nData; }  
            set { nData = value; }  
        }  
  
        public string Sexo  
        {  
            get { return nSexo; }  
            set { nSexo = value; }  
        }  
  
        public string Login  
        {  
            get { return nLogin; }  
            set { nLogin = value; }  
        }  
  
        public int Codigo  
        {  
            get { return nCodigo; }  
            set { nCodigo = value; }  
        }  
  
        public string Email  
        {  
            get { return nEmail; }  
            set { nEmail = value; }  
        }  
  
        public string Senha  
        {  
            get { return nSenha; }  
            set { nSenha = value; }  
        }  
    }  
}
```

```

        public string Nome
        {
            get { return nNome; }
            set { nNome = value; }
        }
    }
}

```

### 2.1.2 Classe Conexao.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using MySql.Data.MySqlClient;

namespace Musitec
{
    //Classe de conexão com o banco.
    class Conexao
    {
        public MySqlConnection cn = new MySqlConnection();

        // Metodo de conexão
        public void conectar()
        {
            cn.ConnectionString = "SERVER = localhost ; DATABASE=Musitec ;UID=root;PASS-
WORD=vertrigo;";
            cn.Open();
        }

        public void desconectar()
        {
            cn.Close();
        }
    }
}

```

### 2.1.3 Classe dal.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using MySql.Data.MySqlClient;

namespace Musitec
{
    class dal
    {
        private MySqlConnection conexao;
    }
}

```

```

public void cadastro(Cadastro cad)
{
    string caminho = "SERVER = localhost ; DATABASE = Musitec ; UID = root; PASSWORD =
vertrigo;";

    try
    {
        conexao = new MySqlConnection(caminho);
        conexao.Open();
        string inserir = " INSERT INTO Cadastro (Login, Email, Senha, NomeCompleto, Sexo, Dat-
aNasc) values ('" + cad.Login + "', '" + cad.Email + "', '" + cad.Senha + "', '" + cad.Nome + "', '" +
cad.Sexo + "', '" + cad.Data + "')";
        MySqlCommand comandos = new MySqlCommand(inserir, conexao);
        comandos.ExecuteNonQuery();
        conexao.Close();
    }

    catch (Exception erro)
    {
        throw new Exception("Erro de comandos!" + erro.Message);
    }
}
}

```

#### 2.1.4 Classe *frmAux1.cs*

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Musitec
{
    public partial class frmAux1 : Form
    {
        int idAula;
        int Modulo;

        public frmAux1(int idAula, int Modulo)
        {
            this.idAula = idAula;
            this.Modulo = Modulo;

            InitializeComponent();
        }

        private void btnVoltar_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        // Formulário auxiliar para puxar os conteudos do banco

```

```

private void frmAux1_Load(object sender, EventArgs e)
{
    Focus();
    Metodos met = new Metodos();
    met.buscarConteudo(idAula, Modulo);
    rtbConteudo.Text = met.NConteudo;
    rtbConteudo.ReadOnly = true;
}
}
}

```

### 2.1.5 Classe frmCadastro.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace Musitec
{
    public partial class frmCadastro : Form
    {
        public frmCadastro()
        {
            InitializeComponent();
            txtNome.Focus();
        }

        private void frmCadastro_Load(object sender, EventArgs e)
        {
            this.MaximizeBox = false;
            this.FormBorderStyle = FormBorderStyle.FixedSingle;
            btnCadastrar.Focus();
        }

        private void btnCadastrar_Click(object sender, EventArgs e)
        {
            if (txtSenha.Text == txtCSenha.Text)
            {
                Cadastro cad = new Cadastro();
                dal da = new dal();
                string dia = DataNasc.Value.Day.ToString();
                string mes = DataNasc.Value.Month.ToString();
                string ano = DataNasc.Value.Year.ToString();
                string data = (ano + "/" + mes + "/" + dia);
                cad.Data = data;
                cad.Nome = txtNome.Text;
                cad.Email = txtEmail.Text;
                cad.Login = txtLogin.Text;
                cad.Senha = txtSenha.Text;

                if (rbtMasculino.Checked == true)
                {

```

```

        cad.Sexo = "M";
    }

    if (rbtFeminino.Checked == true)
    {
        cad.Sexo = "F";
    }

    da.cadastro(cad);
    txtNome.Text = "";
    txtLogin.Text = "";
    txtEmail.Text = "";
    txtSenha.Text = "";
    txtCSenha.Text = "";
    txtNome.Focus();
    MessageBox.Show("Você está registrado!");
    frmLogin log = new frmLogin();
    this.Close();
    log.Show();
}

else
{
    MessageBox.Show("Confirme se sua senha e email estão iguais!");
}
}

//Método que irá voltar à tela de login caso o usuário não se cadastrar.
private void btnVoltar_Click(object sender, EventArgs e)
{
    Form l = new frmLogin();
    this.Close();
    l.Show();
}

private void txtNome_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == 13)
    {
        this.btnCadastrar.PerformClick();
    }
}

private void txtEmail_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == 13)
    {
        this.btnCadastrar.PerformClick();
    }
}

private void rbtMasculino_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == 13)
    {
        this.btnCadastrar.PerformClick();
    }
}

private void rbtFeminino_KeyPress(object sender, KeyPressEventArgs e)

```

```

    {
        if (e.KeyChar == 13)
        {
            this.btnCadastrar.PerformClick();
        }
    }

    private void txtDataNasc_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (e.KeyChar == 13)
        {
            this.btnCadastrar.PerformClick();
        }
    }

    private void txtLogin_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (e.KeyChar == 13)
        {
            this.btnCadastrar.PerformClick();
        }
    }

    private void txtSenha_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (e.KeyChar == 13)
        {
            this.btnCadastrar.PerformClick();
        }
    }

    private void txtCSenha_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (e.KeyChar == 13)
        {
            this.btnCadastrar.PerformClick();
        }
    }
}

```

### 2.1.6 Classe frmLogin.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
namespace Musitec
{
    public partial class frmLogin : Form
    {
        string idLogin;
    }
}

```

```

public frmLogin()
{
    InitializeComponent();
    txtLoginL.TabIndex = 0;
}

private void frmLogin_FormClosed(object sender, FormClosedEventArgs e)
{
    //Application.Exit();
}

private void frmLogin_Load(object sender, EventArgs e)
{
}

private void btnCadastrar_Click(object sender, EventArgs e)
{
    Form c = new frmCadastro();
    this.Hide();
    c.Show();
}

private void btnEntrar_Click(object sender, EventArgs e)
{
    string Login = txtLoginL.Text;
    string Senha = txtSenhaL.Text;
    Conexao conexao = new Conexao();
    conexao.conectar();
    string query = "SELECT Login, Senha From Cadastro WHERE Login = '" + Login + "' AND
Senha = '" + Senha + "'";
    MySqlCommand cmd = new MySqlCommand(query, conexao.cn);
    MySqlDataReader dr = cmd.ExecuteReader();

    if (dr.Read())
    {
        Login = dr["Login"].ToString();
        Senha = dr["Senha"].ToString();
        Metodos met = new Metodos();
        met.buscarId(Login);
        idLogin = met.idLogin;
        Perfil P = new Perfil(idLogin);
        this.Hide();
        P.Show();
    }

    else
    {
        MessageBox.Show("Usuário ou Senha não encontrados!", "Erro!");
    }

    conexao.desconectar();
}

private void txtSenhaL_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == 13)
    {
        this.btnEntrar.PerformClick();
    }
}

```



```

    }

    private void txtLoginL_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (e.KeyChar == 13)
        {
            this.btnEntrar.PerformClick();
        }
    }
}

```

### 2.1.7 Classe metodo.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using MySql.Data.MySqlClient;

```

```

namespace Musitec
{
    class Metodos
    {

```

```

        private string _Nome;
        public string Nome
        {
            get { return _Nome; }
            set { _Nome = value; }
        }

```

```

        private string _Email;
        public string Email
        {
            get { return _Email; }
            set { _Email = value; }
        }

```

```

        private string _idLogin;
        public string idLogin
        {
            get { return _idLogin; }
            set { _idLogin = value; }
        }

```

```

        private string nConteudo;
        public string NConteudo
        {
            get { return nConteudo; }
            set { nConteudo = value; }
        }

```

```

        // Método que busca o conteúdo de acordo com a aula e o modulo por parâmetros passados dos
        // botões de cada modulo e de cada aula
        public void buscarConteudo(int idAula, int Modulo)

```

```

{
    Conexao cx = new Conexao();
    cx.conectar();
    string sql = "";
    sql += " select * from Modulo" + Modulo + " ";
    sql += " where idAula = " + idAula.ToString();
    MySqlCommand cd = new MySqlCommand();
    cd.Connection = cx.cn;
    cd.CommandText = sql;
    MySqlDataReader dr = cd.ExecuteReader();

    if (dr.Read())
    {
        nConteudo = dr["Aula"].ToString();
    }

    dr.Close();
    cx.desconectar();
}

```

//Método que busca o id do usuário para a parte de perfil do programa

```

public void buscarId(string Login)
{
    Conexao cx = new Conexao();
    cx.conectar();
    string sql = "";
    sql = " select idCadastro from Cadastro where Login = @Login";
    MySqlCommand cd = new MySqlCommand();
    cd.Parameters.AddWithValue("@Login", Login);
    cd.Connection = cx.cn;
    cd.CommandText = sql;
    MySqlDataReader dr = cd.ExecuteReader();

    if (dr.Read())
    {
        _idLogin = dr["idCadastro"].ToString();
    }

    dr.Close();
    cx.desconectar();
}

```

// Método que busca o nome e o e-mail do usuário

```

public void buscarNome(string idLogin)
{
    Conexao cx = new Conexao();
    cx.conectar();
    string sql = "";
    sql = " select NomeCompleto,Email from Cadastro where idCadastro = @idLogin";
    MySqlCommand cd = new MySqlCommand();
    cd.Parameters.AddWithValue("@idLogin", idLogin);
    cd.Connection = cx.cn;
    cd.CommandText = sql;
    MySqlDataReader dr = cd.ExecuteReader();

    if (dr.Read())
    {
        _Nome = dr["NomeCompleto"].ToString();
        _Email = dr["Email"].ToString();
    }
}

```

```

        dr.Close();
        cx.desconectar();
    }
}
}

```

### 2.1.8 Classe modulo1.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Drawing.Drawing2D;

namespace Musitec
{
    public partial class Modulo1 : Form
    {
        string idLogin;
        int Modulo = 1;

        public Modulo1(string idLogin)
        {
            this.idLogin = idLogin;
            InitializeComponent();
        }

        private void btnVoltar_Click(object sender, EventArgs e)
        {
            Perfil p = new Perfil(idLogin);
            p.Show();
            this.Close();
        }

        private void btnA1_Click(object sender, EventArgs e)
        {
            int idAula = 1;
            frmAux1 aula = new frmAux1(idAula, Modulo);
            aula.Show();
        }

        private void Modulo1_Load(object sender, EventArgs e)
        {
            Metodos met = new Metodos();
            met.buscarNome(idLogin);
            lblNome.Text = met.Nome;
            lblEmail.Text = met.Email;
        }

        private void btnA2_Click(object sender, EventArgs e)
        {
            int idAula = 2;

```

```
        frmAux1 aula = new frmAux1(idAula, Modulo);
        aula.Show();
    }

    private void btnA3_Click(object sender, EventArgs e)
    {
        int idAula = 3;
        frmAux1 aula = new frmAux1(idAula, Modulo);
        aula.Show();
    }

    private void btnA4_Click(object sender, EventArgs e)
    {
        int idAula = 4;
        frmAux1 aula = new frmAux1(idAula, Modulo);
        aula.Show();
    }

    private void btnA5_Click(object sender, EventArgs e)
    {
        int idAula = 5;
        frmAux1 aula = new frmAux1(idAula, Modulo);
        aula.Show();
    }

    private void btnA6_Click(object sender, EventArgs e)
    {
        int idAula = 6;
        frmAux1 aula = new frmAux1(idAula, Modulo);
        aula.Show();
    }

    private void btnA7_Click(object sender, EventArgs e)
    {
        int idAula = 7;
        frmAux1 aula = new frmAux1(idAula, Modulo);
        aula.Show();
    }

    private void btnA8_Click(object sender, EventArgs e)
    {
        int idAula = 8;
        frmAux1 aula = new frmAux1(idAula, Modulo);
        aula.Show();
    }

    private void btnA9_Click(object sender, EventArgs e)
    {
        int idAula = 9;
        frmAux1 aula = new frmAux1(idAula, Modulo);
        aula.Show();
    }
}
```

```

private void btnA10_Click(object sender, EventArgs e)
{
    int idAula = 10;
    frmAux1 aula = new frmAux1(idAula, Modulo);
    aula.Show();
}

private void btnA11_Click(object sender, EventArgs e)
{
    int idAula = 11;
    frmAux1 aula = new frmAux1(idAula, Modulo);
    aula.Show();
}

private void btnA12_Click(object sender, EventArgs e)
{
    int idAula = 12;
    frmAux1 aula = new frmAux1(idAula, Modulo);
    aula.Show();
}

private void btnA13_Click(object sender, EventArgs e)
{
    int idAula = 13;
    frmAux1 aula = new frmAux1(idAula, Modulo);
    aula.Show();
}

private void btnA14_Click(object sender, EventArgs e)
{
    int idAula = 14;
    frmAux1 aula = new frmAux1(idAula, Modulo);
    aula.Show();
}

private void btnA15_Click(object sender, EventArgs e)
{
    int idAula = 15;
    frmAux1 aula = new frmAux1(idAula, Modulo);
    aula.Show();
}

private void btnA16_Click(object sender, EventArgs e)
{
    int idAula = 16;
    frmAux1 aula = new frmAux1(idAula, Modulo);
    aula.Show();
}
}
}

```

A classe *modulo1.cs* estende-se até o módulo 16, sendo assim temos 16 classes

(*modulo1.cs, modulo2.cs, modulo3.cs... até modulo16.cs*), decidimos optar por este método pois em cada módulo, que está dividido por aulas, contém quantidades de imagens diferentes, então seria necessário desativar as *picturebox's*, em cada botão inserido, dificultando a lógica para mostrar o conteúdo com a devida formatação.

## 2.2 Tecnologias empregadas

Desenvolvemos o projeto utilizando, para o programa, que é a parte central do projeto, a linguagem de programação C#, compatível com o sistema operacional Windows. Esta foi a linguagem escolhida pois o Windows é o sistema operacional que abrange a maior parte dos computadores e notebooks vendidos e usados (para uso pessoal), então o problema de compatibilidade hoje não seria uma desculpa muito usada para o nosso projeto não rodar em máquinas e para quem deseja aprender e se desenvolver quando se fala em teoria musical.

Outro ponto foi que a nossa equipe tem mais afinidade nesta linguagem, o C#, logo seria mais fácil e viável. A história do C#

A linguagem C# foi criada junto com a arquitetura .NET. Embora existam várias outras linguagens que suportam essa tecnologia (como VB.NET, C++, J#), C# é considerada a linguagem símbolo do .NET pelas seguintes razões:

- Foi criada praticamente do zero para funcionar na nova plataforma, sem preocupações de compatibilidade com código de legado.
- O compilador C# foi o primeiro a ser desenvolvido.
- A maior parte das classes do .NET Framework foram desenvolvidas em C#.

A criação da linguagem, embora tenha sido feita por vários desenvolvedores, é atribuída principalmente a Anders Hejlsberg, hoje um Distinguished Engineer na Microsoft. Anders Hejlsberg era desenvolvedor de compiladores na Borland, e entre suas criações mais conhecidas estão o Turbo Pascal e o Delphi.

### *2.2.1 Criação do nome*

Muitos pensam que o nome C# viria de uma sobreposição de 4 símbolos "+" dando a impressão de "++++". Na verdade, o "#" de C# refere-se ao sinal musical (sustenido), que aumenta em 1/2 tom uma nota musical. O símbolo real seria o  $\sharp$  e não o #, porém, devido a limitação de telas, fontes e alguns browsers, no momento da normalização junto a ECMA, foi especificado apenas que o nome da linguagem seria uma letra C maiúscula (U+0043) e o sinal "#" (U+0023), facilitando assim, publicações e artigos com um carácter encontrado facilmente dos layouts de teclado padrões. Desta forma, caso o nome fosse usado em português, seria "C-Sustenido" (ou "Dó-Sustenido"), e não "C-cerquilha".! para fazermos um projeto de qualidade.

### 3 MODELAGEM LÓGICA

A modelagem lógica é segunda etapa da modelagem de dados, que se refere à forma conjunta que possibilita a interconexão prática do projeto. Nesse modelo serão definidos os padrões e nomenclaturas, chaves primárias e estrangeiras, sempre levando em conta o modelo conceitual criado anteriormente.

Definindo essa modelagem, ela é a responsável por obter o levantamento de dados que darão suporte à construção de todo o modelo.

#### 3.1 Diagrama Entidade Relacionamento

No diagrama abaixo vemos a comunicação de cada tabela no banco de dados da nossa aplicação, onde cada tabela adquirirá informações de outra ou até mesmo incluir informações em determinada tabela. O Diagrama Entidade Relacionamento é a representação gráfica do Modelo Entidade Relacionamento, facilitando o entendimento sobre o sistema e seu desenvolvimento.

O Usuário pode efetuar um Registro ou efetuar o Login caso já tenha registrado. Cada Login possui uma introdução básica do aplicativo, logo após isso o Usuário escolherá seus instrumentos e seus níveis de aprendizagem, cada nível possui um conteúdo diferente.

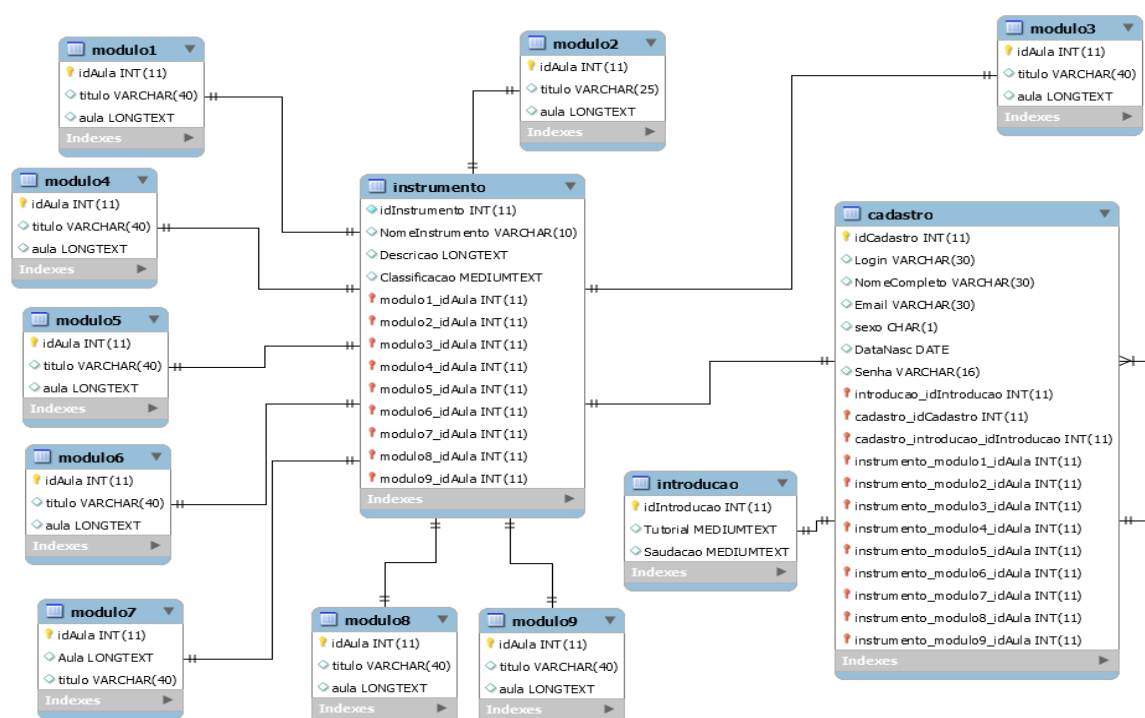


Figura 6 - Diagrama Entidade Relacionamento do banco de dados



### 3.1.1 Relacionamentos

- **Usuário efetua o cadastro:** O usuário pode efetuar diversos cadastros;
- **Usuário efetua Login:** O usuário pode efetuar um login e cada login possui somente um usuário;
- **Login contém Introdução:** Cada login (utilizando as variáveis do cadastro) possui uma introdução;
- **Instrumento possui Nível:** Cada instrumento possui vários níveis e vários níveis possuem um instrumento;
- **Nível contém Conteúdo:** Um nível (módulo) possui vários conteúdos (as Aulas) e os conteúdos possuem apenas um nível.

### 3.1.2 Diagrama de Caso de Uso

O Diagrama de Caso de Uso mostra a aplicação do ponto de vista do usuário. Ele mostra todas as opções de interação disponíveis para o usuário, as decisões que ele precisa tomar e os respectivos resultados com base em suas decisões dentro da aplicação. O diagrama é responsável por uma melhor explicação visual de como o usuário enxerga as funções da aplicação

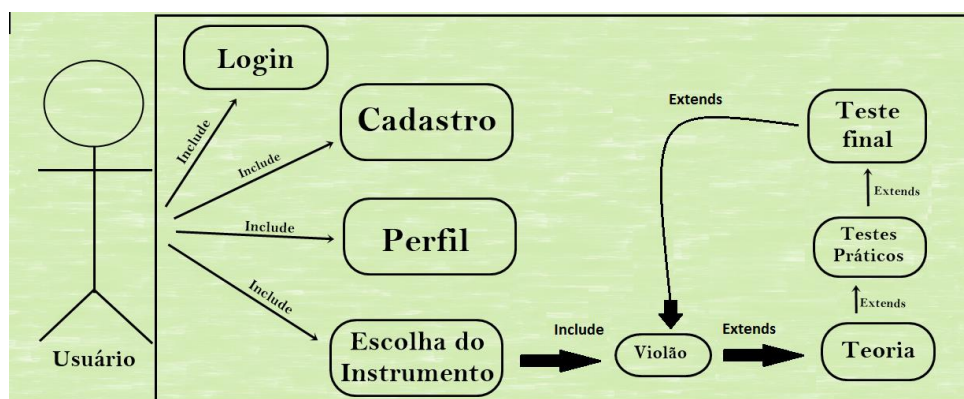


Figura 7 - Caso de Uso

## 3.2 Especificação de Caso de Uso

Nessa especificação de caso de uso descreveremos uma sequência de ações com o objetivo de mostrar como o sistema se comportará quando houver

interação com o usuário. No caso só teremos uma tabela que será a de login, onde temos uma ilustração abaixo especificando como o sistema conversará com o usuário.

*Tabela 1 - Cadastro*

Caso de Uso	Cadastro
Caso de Uso Geral	Incluir, alterar, inativar e consultar
Ator Principal	Usuário
Pré-condição	-----
Fluxo Normal	Cadastrar nome, senha e e-mail
Fluxos Excepcionais	Confirmar Cadastro
Fluxos Alternativos	Entrar, pois já há cadastro
Pós-condição	Cadastro efetuado com sucesso

*Tabela 2 - Login*

Caso de Uso	Login
Caso de Uso Geral	Incluir, alterar, inativar e consultar
Ator Principal	Usuário
Pré-condição	-----
Fluxo Normal	Entrar com o e-mail e a senha
Fluxos Excepcionais	Efetuar o Login
Fluxos Alternativos	-----
Pós-condição	Usuário Logado

*Tabela 3 - Perfil*

Caso de Uso	Perfil
Caso de Uso Geral	Incluir, alterar, inativar e consultar
Ator Principal	Usuário

Pré-condição	Estar logado
Fluxo Normal	Verificar o perfil e o progresso
Fluxos Excepcionais	Escolher um instrumento musical
Fluxos Alternativos	Escolher um nível para aprender
Pós-condição	Ir para escolher instrumento

*Tabela 4 - Instrumento*

Caso de Uso	Instrumento
Caso de Uso Geral	Incluir, alterar, inativar e consultar
Ator Principal	Usuário
Pré-condição	Estar logado
Fluxo Normal	Níveis do Violão
Fluxos Excepcionais	Nível já concluído
Fluxos Alternativos	Você não chegou nesse nível ainda
Pós-condição	Nível

*Tabela 5 - Teoria*

Caso de Uso	Teoria
Caso de Uso Geral	Incluir, alterar, inativar e consultar
Ator Principal	Usuário
Pré-condição	Estar logado
Fluxo Normal	Conteúdo em teoria no nível previamente selecionado
Fluxos Excepcionais	Nível concluído
Fluxos Alternativos	Nível não concluído
Pós-condição	Nível concluído com sucesso

Caso de Uso	Teste prático (Exercícios)
Caso de Uso Geral	Incluir, alterar, inativar e consultar
Ator Principal	Usuário
Pré-condição	Estar logado
Fluxo Normal	Ouvir e responder
Fluxos Excepcionais	Teste concluído
Fluxos Alternativos	Teste não concluído
Pós-condição	Teste concluído com sucesso

*Tabela 6 - Teste final*

Caso de Uso	Teste final
Caso de Uso Geral	Incluir, alterar, inativar e consultar
Ator Principal	Usuário
Pré-condição	Estar logado e ter feito a teoria e os testes práticos
Fluxo Normal	Prova final
Fluxos Excepcionais	Teste final concluído, ir para o próximo nível
Fluxos Alternativos	Teste final não concluído, refazer
Pós-condição	Ir para o próximo nível

## 4 NOTAS DE VERSÃO

Aqui é mostrado as notas de versão mais importantes durante o desenvolvimento do projeto Musitec, enfatizando a parte de desenvolvimento do programa.

Musitec v1.0 (15/04/2016)

- Criado os formulários;
- Definido a quantidade de módulos;
- Criado a versão inicial do banco de dados.

Musitec v1.1 (06/05/2016)

- Adicionado os botões dos formulários feitos;
- Adicionado novas tabelas no banco de dados;

Musitec v1.3 (20/05/2016)

- Removido alguns botões nos formulários;
- Feito o esboço do layout;
- Banco de dados modificado devido a quantidade de módulos e aulas terem sido alteradas;

Musitec v1.7 (10/06/2016)

- Adicionado o layout no projeto Musitec;
- Adicionado tratamentos de erro nos formulários cadastro.cs e login.cs;
- Adicionado parte do conteúdo no banco de dados.

Musitec v1.7.2 (24/6/2016)

- Removido conteúdos do banco de dados.

Musitec v2.0 (05/06/2016)

- Banco de dados remodelado;
- Conteúdos adicionados novamente.

Musitec v2.4 (19/06/2016)

- Layout do projeto remodelado;
- Lógica dos códigos refeita.

Musitec v2.4.5 (26/08/2016)

- Código da aplicação melhorado (cadastro e login).

Musitec v2.6 (02/09/2016)

- Lógica dos conteúdos melhorada, diminuindo as linhas e métodos necessários.

Musitec v3.0 (18/11/2016)

- Adicionado todo o conteúdo no banco de dados;
- Feito todos os tratamentos de erro necessários para um bom uso no projeto.
- Adicionado o layout do projeto.

## CONCLUSÃO

Até o então momento – Novembro de 2016 – expusemos o que e como as nossas implementações puderam ser feitas. Percebe-se que as tecnologias mudaram bruscamente para uma melhora na vida das pessoas, foi escolhido o C# como linguagem de programação. Pois é uma linguagem onde possui uma IDE (Área de trabalho) intuitiva e fácil de usar, assim como também facilita para os usuários, visto que grande parte da população usa, atualmente, o Windows como sistema operacional.

Não houve problemas com relação a um setor do sistema entrar em conflito com o outro. Salientamos, no entanto, que o que fora desenvolvido até o momento ainda não vale como definitivo, pois, como sabemos, a apresentação é parcial, e o projeto é o final, assim como será a documentação.

Os próximos passos a serem adotados são uma melhor utilização dos recursos expostos e utilizados neste projeto, bem como revisão das documentações e das tecnologias.

## REFERÊNCIAS

Med, Bohumil. ***Teoria da Música***. 4ª. Edição, São Paulo

Penna, Maura. Construindo o primeiro projeto de pesquisa em Educação e Música. 1ª Edição, São Paulo, 2015.

Penna, Maura. ***Música(s) e seu ensino***. 2ª Edição, São Paulo, 2015

A história do C#, acessado em 18/09;2016:

<https://www.portaleducacao.com.br/informatica/artigos/6137/historia-e-caracteristicas-da-linguagem-c>