

Vitor Bruno de Oliveira Barth

Sistema de Controle de Horários e Matrículas para Escolas de Música

Cuiabá
2017

Nome do Sistema: Sistema de Controle de Horários e Matrículas para Escolas de Música

Objetivo: Controlar os horários de aula, pagamento a professores e de alunos de uma escola de música.

Justificativa:

Controlar uma escola de música com diversos professores e alta rotatividade de professores é algo complexo, o que pode ocasionar problemas que trazem prejuízos à escola e aos alunos, como cobranças ou pagamentos indevidos, presença ou ausência inesperada de alunos, aulas em excesso ou em falta, etc.

O desenvolvimento deste sistema permitirá ao administrador da escola evitar essas falhas através da organização do uso do tempo dos professores e do controle sobre as despesas e receitas da escola.

Funcionalidades:

- a) Cadastro de Professores
- b) Cadastro de Alunos
- c) Agenda de Aulas
- d) Registro de Recebimentos
- e) Registro de Pagamentos

Recursos Utilizados

- Hardware:
 - Computador
- Software:
 - Banco de Dados: PostgreSQL
 - Servidor Web: Node.js
 - Linguagem de Marcação: Pug.js
 - Linguagem de Folha de Estilos: CSS

Diagrama Entidade-Relacionamento Conceitual

Proposta de Banco de Dados
Escola de Música

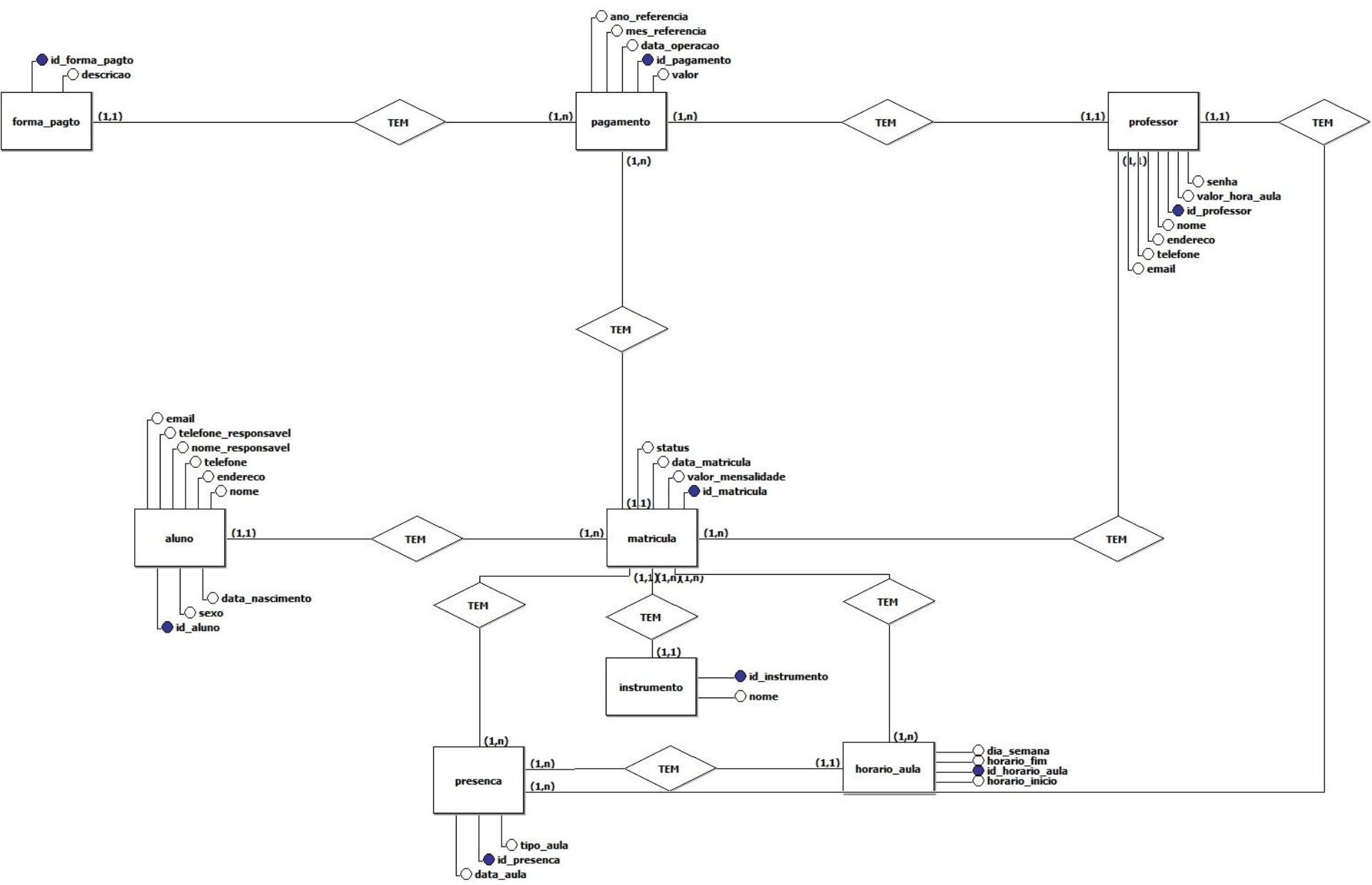
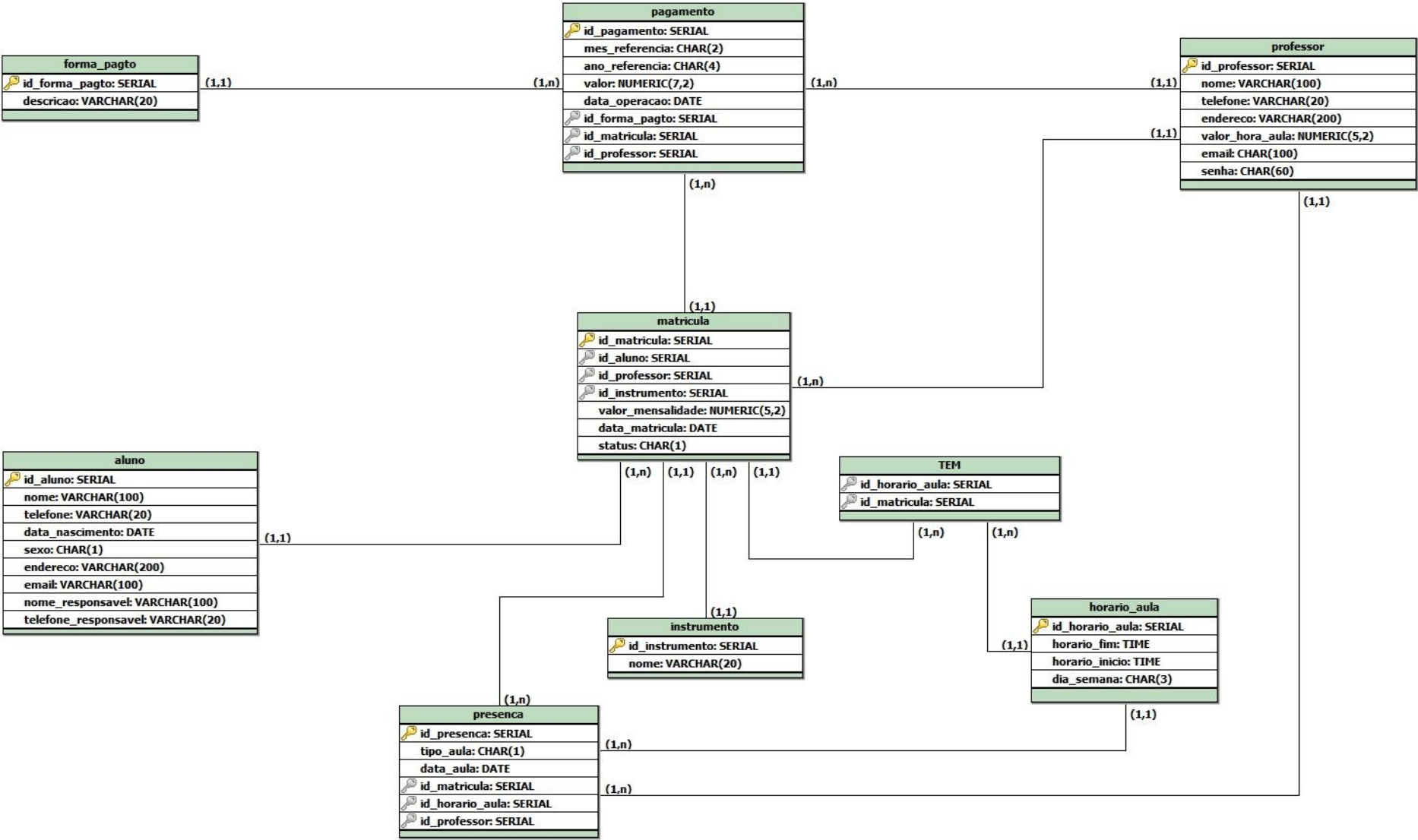


Diagrama Entidade-Relacionamento Lógico

Proposta de Banco de Dados
Escola de Música



Dicionário de Dados

- aluno

Atributo	Tipo	Obrigatório	Descrição
id_aluno	SERIAL	S	Chave Primária
nome_aluno	VARCHAR(100)	S	Nome do Aluno
telefone	VARCHAR(20)	S	Telefone Pessoal do Aluno
nome_responsavel	CHAR(100)	N	Nome do Responsável, caso o Aluno seja Menor de Idade
telefone_responsavel	VARCHAR(20)	N	Telefone do Responsável, caso o Aluno seja Menor de Idade
email	VARCHAR(100)	N	E-Mail Pessoal do Aluno
endereço	VARCHAR(200)	S	Endereço Residencial do Aluno
sexo	CHAR(1)	S	Sexo do Aluno
data_nascimento	DATE	S	Data de Nascimento do Aluno

- professor

Atributo	Tipo	Obrigatório	Descrição
id_professor	SERIAL	S	Chave Primária
nome	VARCHAR(100)	S	Nome do Professor
telefone	VARCHAR(20)	S	Telefone Pessoal do Professor
valor_hora_aula	NUMERIC(5,2)	S	Valor da Hora de Aula Cobrado
endereco	VARCHAR(200)	S	Endereço Residencial do Professor
email	VARCHAR(100)	S	E-mail do Professor Utilizado para Acesso ao Sistema
senha	CHAR(60)	S	Senha de Acesso ao Sistema

- instrumento

Atributo	Tipo	Obrigatório	Descrição
id_instrumento	SERIAL	S	Chave Primária
nome	VARCHAR(20)	S	Nome do Instrumento

- matricula

Atributo	Tipo	Obrigatório	Descrição
id_matricula	SERIAL	S	Chave Primária
valor_mensalidade	NUMERIC(5,2)	S	Valor da Mensalidade Atual
status	CHAR(1)	S	Indica o Status da Matrícula. A → Ativo; I → Inativo;
data_matricula	DATE	S	Data de Realização da Matrícula
id_instrumento	SERIAL	S	Chave Estrangeira Instrumento Estudado
id_aluno	SERIAL	S	Chave Estrangeira Aluno ao qual a matrícula pertence
id_professor	SERIAL	S	Chave Estrangeira Professor que ministra a aula
id_horario_aula	SERIAL	S	Chave Estrangeira Horário de Aula

- horario_aula

Atributo	Tipo	Obrigatório	Descrição
id_horario_aula	SERIAL	S	Chave Primária
horario_inicio	TIME	S	Horário de Início da Aula
horario_fim	TIME	S	Horário de Fim da Aula
dia_semana	CHAR(3)	S	Dia da Semana SEG → segunda-feira TER → terça-feira QUA → quarta-feira QUI → quinta-feira SEX → sexta-feira SAB → sábado

- presenca

Atributo	Tipo	Obrigatório	Descrição
id_presenca	SERIAL	S	Chave Primária
tipo_aula	CHAR(1)	S	Tipo de Aula N → Normal R → Reposição E → Extra
data_aula	DATE	S	Data que a aula foi realizada
id_matricula	SERIAL	S	Chave Estrangeira Matrícula a qual se refere a aula
id_horario_aula	SERIAL	S	Horário em que a aula foi realizada
id_professor	SERIAL	S	Professor que ministrou a aula

- pagamentos

Atributo	Tipo	Obrigatório	Descrição
id_pagamento	SERIAL	S	Chave Primária
mes_referencia	CHAR(2)	S	Mês de Referência
ano_referencia	CHAR(4)	S	Ano de Referência
data_operacao	DATE	S	Data de Realização da Operação
valor	NUMERIC(7,2)	S	Valor da Operação
id_forma_pagto	SERIAL	S	Chave Estrangeira Forma de Pagamento Utilizada
id_professor	SERIAL	N	Chave Estrangeira Professor ao qual o pagamento se destina.
id_matricula	SERIAL	N	Chave Estrangeira Matrícula a qual se refere o pagamento

- forma_pagto

Atributo	Tipo	Obrigatório	Descrição
id_forma_pagto	SERIAL	S	Chave Primária
descricao	VARCHAR(20)	S	Descrição da Forma de Pagamento

Script de Criação do Banco de Dados

```
CREATE SCHEMA escola_musica;
```

```
USE escola_musica;
```

```
CREATE TABLE forma_pagto (  
    id_forma_pagto SERIAL PRIMARY KEY,  
    descricao VARCHAR(20)  
);
```

```
CREATE TABLE instrumento (  
    id_instrumento SERIAL PRIMARY KEY,  
    nome VARCHAR(20)  
);
```

```
CREATE TABLE professor (  
    id_professor SERIAL PRIMARY KEY,  
    nome VARCHAR(100),  
    telefone VARCHAR(20),  
    endereco VARCHAR(200),  
    valor_hora_aula NUMERIC(5,2),  
    email CHAR(100),  
    senha CHAR(60)  
);
```

```
CREATE TABLE aluno (  
    id_aluno SERIAL PRIMARY KEY,  
    nome VARCHAR(100),  
    telefone VARCHAR(20),  
    data_nascimento DATE,  
    sexo CHAR(1),  
    endereco VARCHAR(200),  
    email VARCHAR(100),  
    nome_responsavel VARCHAR(100),  
    telefone_responsavel VARCHAR(20)  
);
```

```
CREATE TABLE matricula (  
    id_matricula SERIAL PRIMARY KEY,  
    id_aluno BIGINT UNSIGNED,  
    id_professor BIGINT UNSIGNED,  
    id_instrumento BIGINT UNSIGNED,  
    valor_mensalidade NUMERIC(5,2),  
    data_matricula DATE,  
    status CHAR(1),  
    FOREIGN KEY(id_aluno) REFERENCES aluno (id_aluno),
```

```
FOREIGN KEY(id_professor) REFERENCES professor (id_professor),  
FOREIGN KEY(id_instrumento) REFERENCES instrumento (id_instrumento)  
);
```

```
CREATE TABLE horario_aula (  
    id_horario_aula SERIAL PRIMARY KEY,  
    horario_fim TIME,  
    horario_inicio TIME,  
    dia_semana CHAR(3)  
);
```

```
CREATE TABLE matricula_horario_aula (  
    id_horario_aula BIGINT UNSIGNED,  
    id_matricula BIGINT UNSIGNED,  
    FOREIGN KEY(id_horario_aula) REFERENCES horario_aula (id_horario_aula),  
    FOREIGN KEY(id_matricula) REFERENCES matricula (id_matricula)  
);
```

```
CREATE TABLE presenca (  
    id_presenca SERIAL PRIMARY KEY,  
    tipo_aula CHAR(1),  
    data_aula DATE,  
    id_matricula BIGINT UNSIGNED,  
    id_horario_aula BIGINT UNSIGNED,  
    id_professor BIGINT UNSIGNED,  
    FOREIGN KEY(id_matricula) REFERENCES matricula (id_matricula),  
    FOREIGN KEY(id_professor) REFERENCES professor (id_professor),  
    FOREIGN KEY(id_horario_aula) REFERENCES horario_aula (id_horario_aula)  
);
```

```
CREATE TABLE pagamento (  
    id_pagamento SERIAL PRIMARY KEY,  
    mes_referencia CHAR(2),  
    ano_referencia CHAR(4),  
    valor NUMERIC(7,2),  
    data_operacao DATE,  
    id_forma_pagto BIGINT UNSIGNED,  
    id_matricula BIGINT UNSIGNED,  
    id_professor BIGINT UNSIGNED,  
    FOREIGN KEY(id_forma_pagto) REFERENCES forma_pagto (id_forma_pagto),  
    FOREIGN KEY(id_matricula) REFERENCES matricula (id_matricula),  
    FOREIGN KEY(id_professor) REFERENCES professor (id_professor)  
);
```

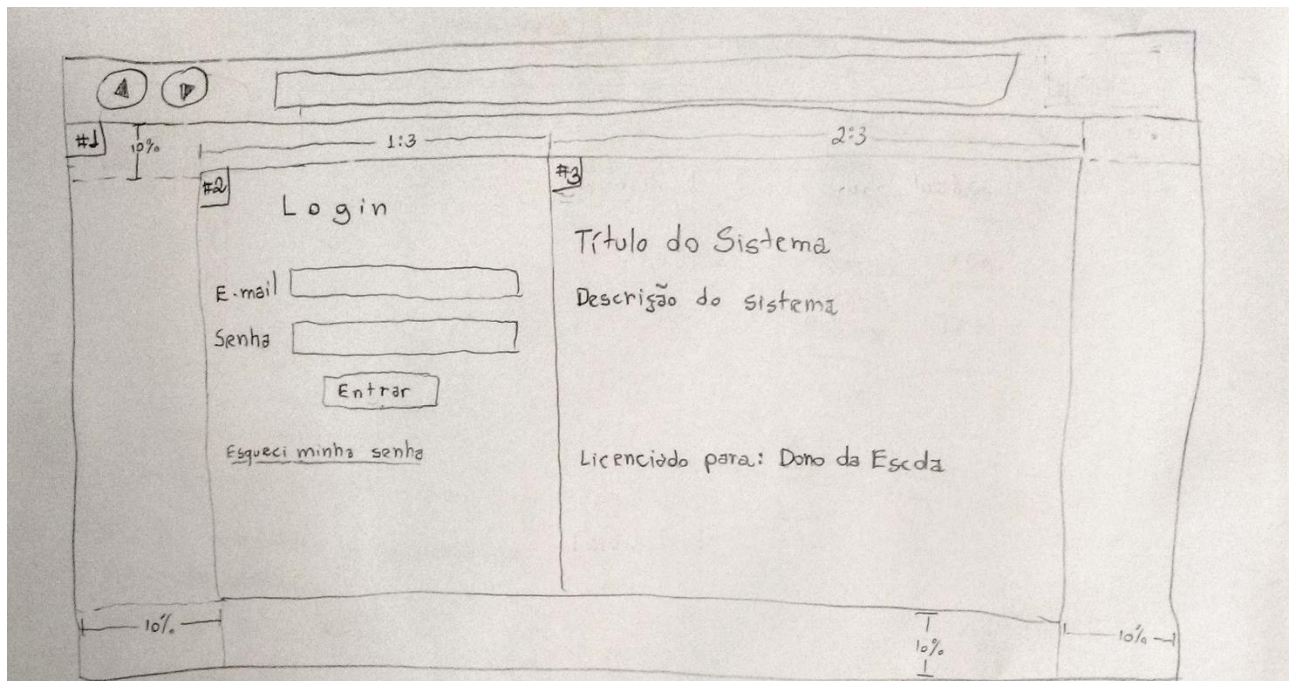
Cronograma (Atualizado)

Tarefa	Início		Fim		Situação	Observações
	Previsto	Realizado	Previsto	Realizado		
Modelagem	21/09/2017	21/09/2017	30/11/2017	28/11/2017	Concluído	
Proposta de Layout	22/11/2017	26/11/2017	30/11/2017	29/11/2017	Concluído	
Dicionário de Dados	29/11/2017	26/11/2017	30/11/2017	29/11/2017	Concluído	
Criação do Banco de Dados	30/11/2017	30/11/2017	30/11/2017	30/11/2017	Concluído	
Conexão com o B.D.	01/12/2017	30/11/2017	02/12/2017	30/11/2017	Concluído	
Design da Tela de Login	01/12/2017	04/12/2017	02/12/2017	06/12/2017	Concluído	
Funcionalidade de Login	04/12/2017	08/12/2017	05/12/2017	10/12/2017	Concluído	
Design da Tela Principal	06/12/2017	10/12/2017	11/12/2017		50%	Foi criada a tela principal, contudo o Design não está concluído.
Design das Telas de Cadastro de Professores e Alunos	12/12/2017		14/12/2017			
Funcionalidades de Cadastro de Professores e Alunos	15/12/2017		18/12/2017			
Design das Telas de Cadastro de Matrículas e Horários	18/12/2017		20/12/2017			
Funcionalidades de Cadastro de Matrículas e Horários	21/12/2017		23/12/2017			
Design das Telas de Cadastro de Pagamentos	08/01/2018		12/01/2018			

Funcionalidades das Telas de Cadastro de Pagamentos	15/01/2018		19/01/2018			
Desenvolvimento do Manual	22/01/2018		26/01/2018			
Testes e Otimização	30/01/2018		06/02/2018			
Implantação	07/02/2017		13/02/2018			

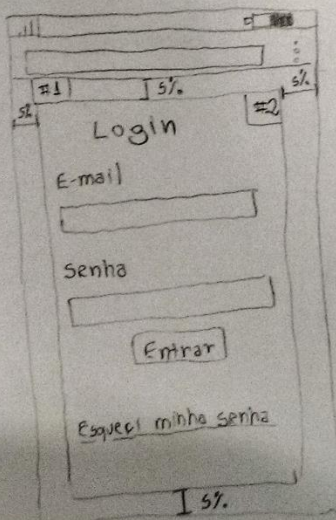
Proposta de Layout

Tela de Login



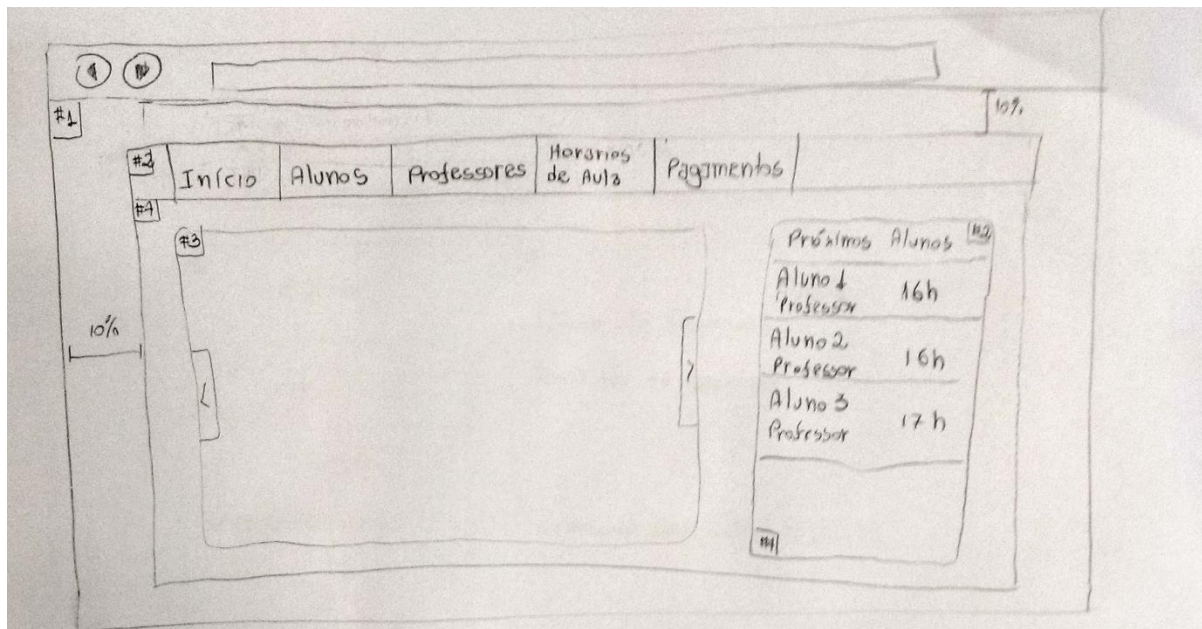
Planos de Fundo:

- #1 - Cor sólida ou imagem; Esmaecida
- #2 - Branco ou cor muito clara
- #3 - Cor sólida ou imagem; Esmaecida

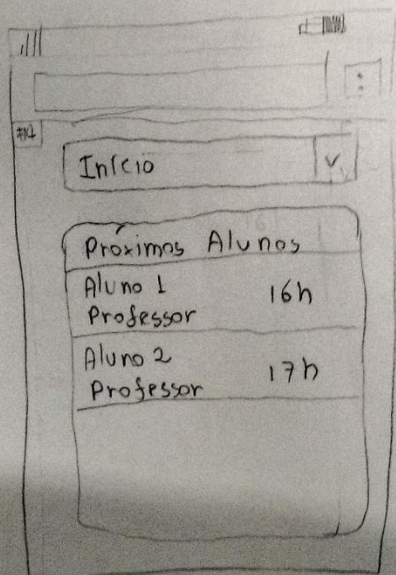


Vitor Bruno de O. Borin

Tela Principal



- #1 - Janela principal com todos os elementos
- #2 - Lista de alunos
- #3 - Cursor de texto
- #4 - Com todos os dados



Tela de Cadastro

The sketch shows a web interface for a registration system. At the top, there is a header bar with two circular icons (a left arrow and a right arrow) and a long rectangular input field. Below this is a navigation menu with five tabs: 'Início', 'Alunos' (which is highlighted with diagonal hatching), 'Professores', 'Horários de Aula', and 'Pagamentos'. Under the 'Alunos' tab, there is a sub-menu with three options: 'Cadastro' (highlighted with a thick border), 'Listagem', and 'Matrículas'. The main content area is titled 'Cadastrar Aluno'. It contains several form fields: 'Nome' (a long horizontal box), 'Telefone' (a horizontal box), 'Endereço' (a long horizontal box), 'Sexo' (with radio buttons for 'Masculino' and 'Feminino'), 'E-mail' (a horizontal box), 'Data de Nascimento' (a horizontal box), 'Nome do Responsável' (a horizontal box), and 'Telefone do Responsável' (a horizontal box). Below these fields is a section labeled 'Matrículas:' which contains a table. The table has two columns: the first column contains the text 'Violão' and the second column contains 'terça 16h-17h'. Below the table is a button labeled 'Adicionar'. At the bottom right of the form is a button labeled 'Gravar'.

Início	Alunos	Professores	Horários de Aula	Pagamentos
Cadastro	Listagem	Matrículas		

Cadastrar Aluno

Nome

Telefone

Endereço

Sexo ☐ Masculino ☐ Feminino

E-mail

Data de Nascimento

Nome do Responsável

Telefone do Responsável

Matrículas:

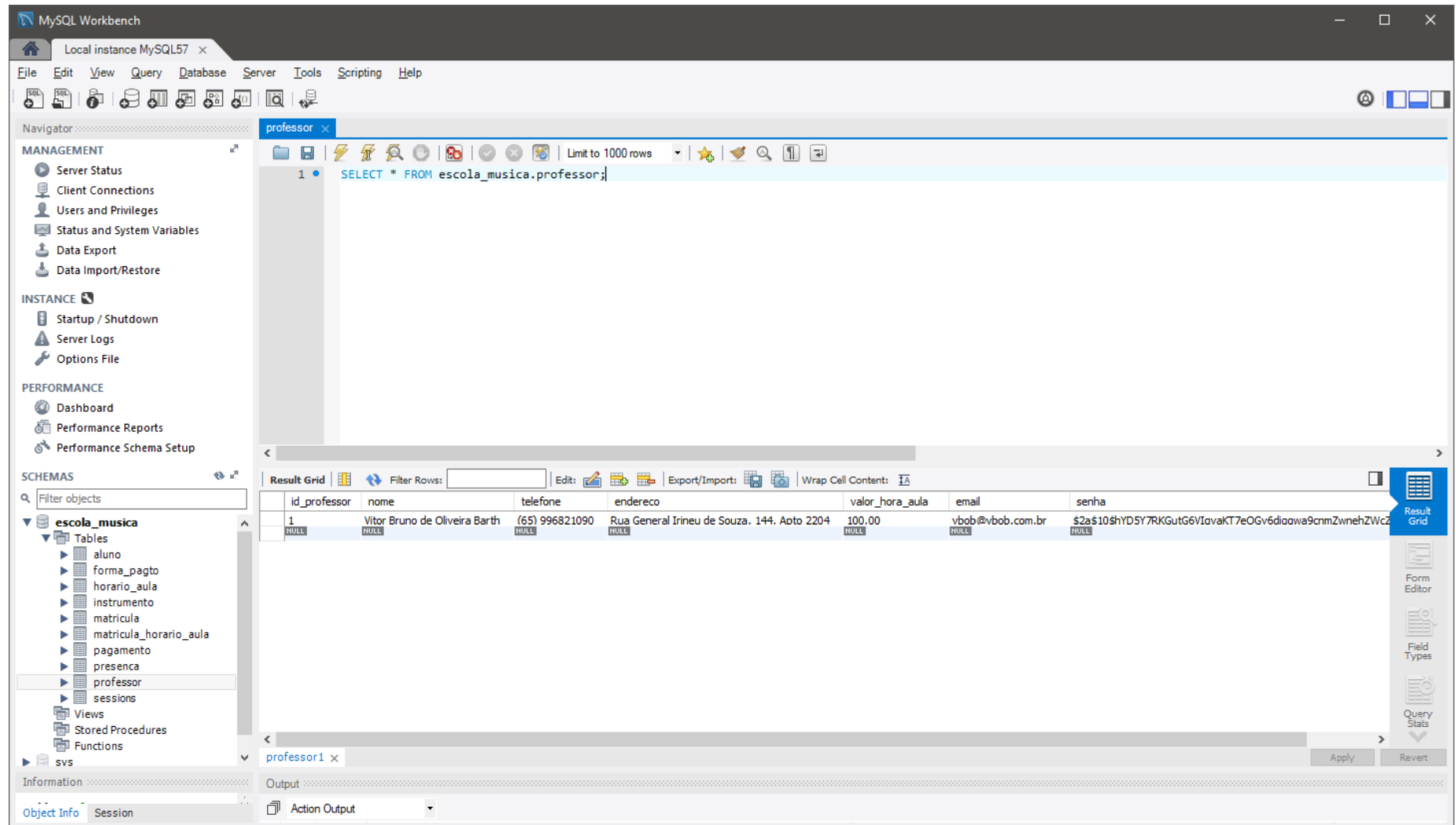
Violão	terça 16h-17h
--------	---------------

Adicionar

Gravar

Andamento 27/11/2017 a 10/12/2017

Criação do Banco de Dados: O Banco de Dados foi gerado, e foi criada a primeira entrada, com um login em senha. Essa entrada foi criada manualmente, usando script SQL.



The screenshot shows the MySQL Workbench interface. The left sidebar contains the 'MANAGEMENT' section with options like Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, and Data Import/Restore. Below this is the 'INSTANCE' section with Startup / Shutdown, Server Logs, and Options File. The 'PERFORMANCE' section includes Dashboard, Performance Reports, and Performance Schema Setup. The 'SCHEMAS' section shows a tree view of the database structure, with 'escola_musica' expanded to show tables like aluno, forma_pagto, horario_aula, instrumento, matricula, matricula_horario_aula, pagamento, presenca, professor, sessions, Views, Stored Procedures, Functions, and svcs.

The main query editor shows the following SQL query:

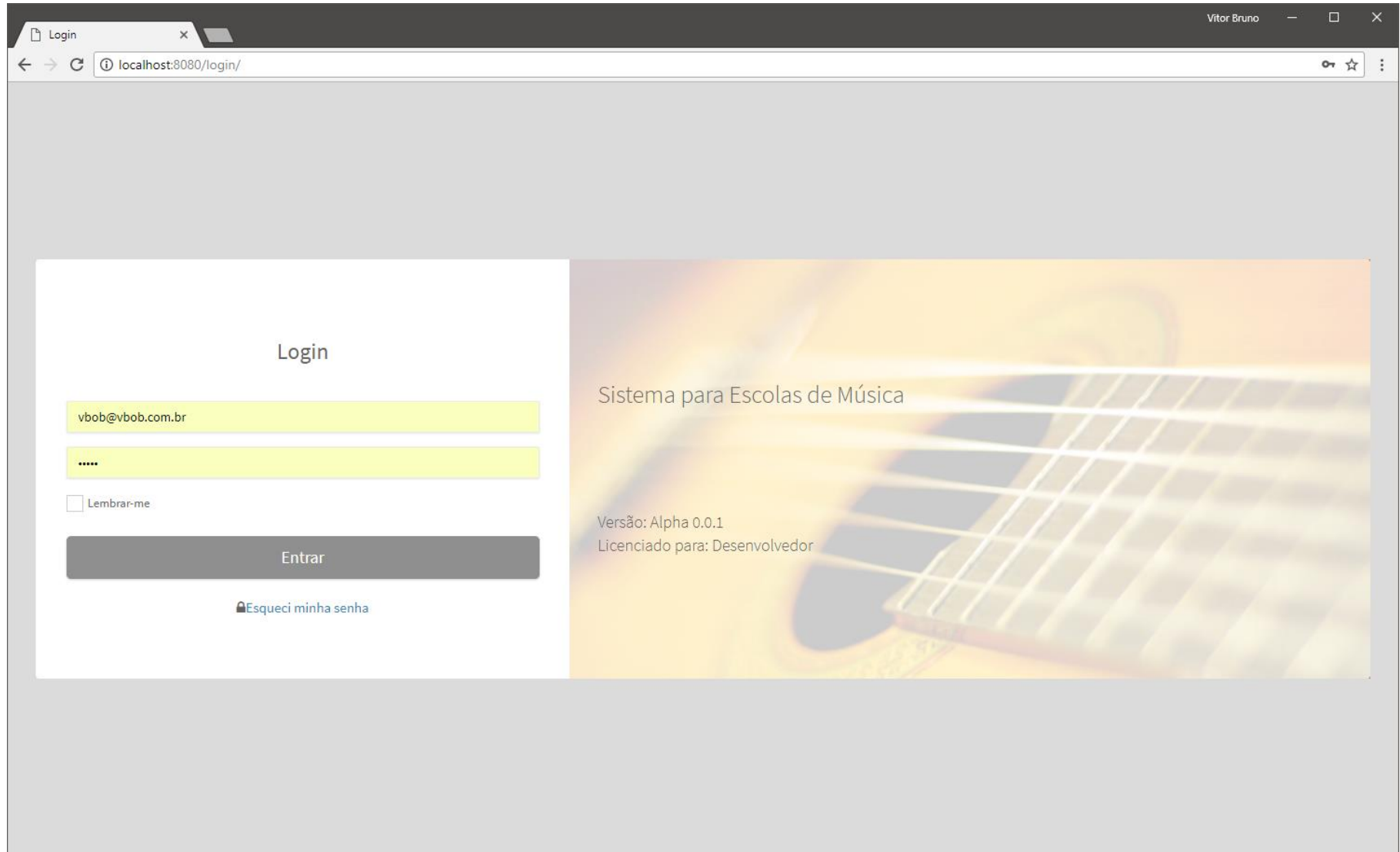
```
SELECT * FROM escola_musica.professor;
```

The query results are displayed in a table with the following columns: id_professor, nome, telefone, endereco, valor_hora_aula, email, and senha. The results show one record for a professor named Vitor Bruno de Oliveira Barth.

id_professor	nome	telefone	endereco	valor_hora_aula	email	senha
1	Vitor Bruno de Oliveira Barth	(65) 996821090	Rua General Irineu de Souza. 144. Apto 2204	100.00	vbob@vbob.com.br	\$2a\$10\$yD5Y7RKutG6VZavaKT7eOGv6diaowa9cnmZwnehZWcz

The bottom of the interface shows the 'Output' tab with 'Action Output' selected.

Tela de Login: Foi criada a tela de login, e suas funcionalidades (exibir mensagens de erros, autenticação de usuário e armazenamento de cookies).



The screenshot shows a web browser window with the title 'Login' and the address bar displaying 'localhost:8080/login/'. The page content is divided into two main sections. On the left, there is a white login form with the title 'Login'. It contains two input fields: the first is filled with 'vbob@vbob.com.br' and the second is masked with '*****'. Below these fields is a checkbox labeled 'Lembrar-me' and a dark gray button labeled 'Entrar'. At the bottom of the form is a link that says 'Esqueci minha senha' with a small lock icon. On the right, there is a large banner image with a warm, orange-toned background showing a close-up of a guitar fretboard. Overlaid on this image is the text 'Sistema para Escolas de Música' in a large, white font. Below this, in a smaller white font, are the details 'Versão: Alpha 0.0.1' and 'Licenciado para: Desenvolvedor'.

Browser: Login
Address: localhost:8080/login/

Login

vbob@vbob.com.br

☐ Lembrar-me

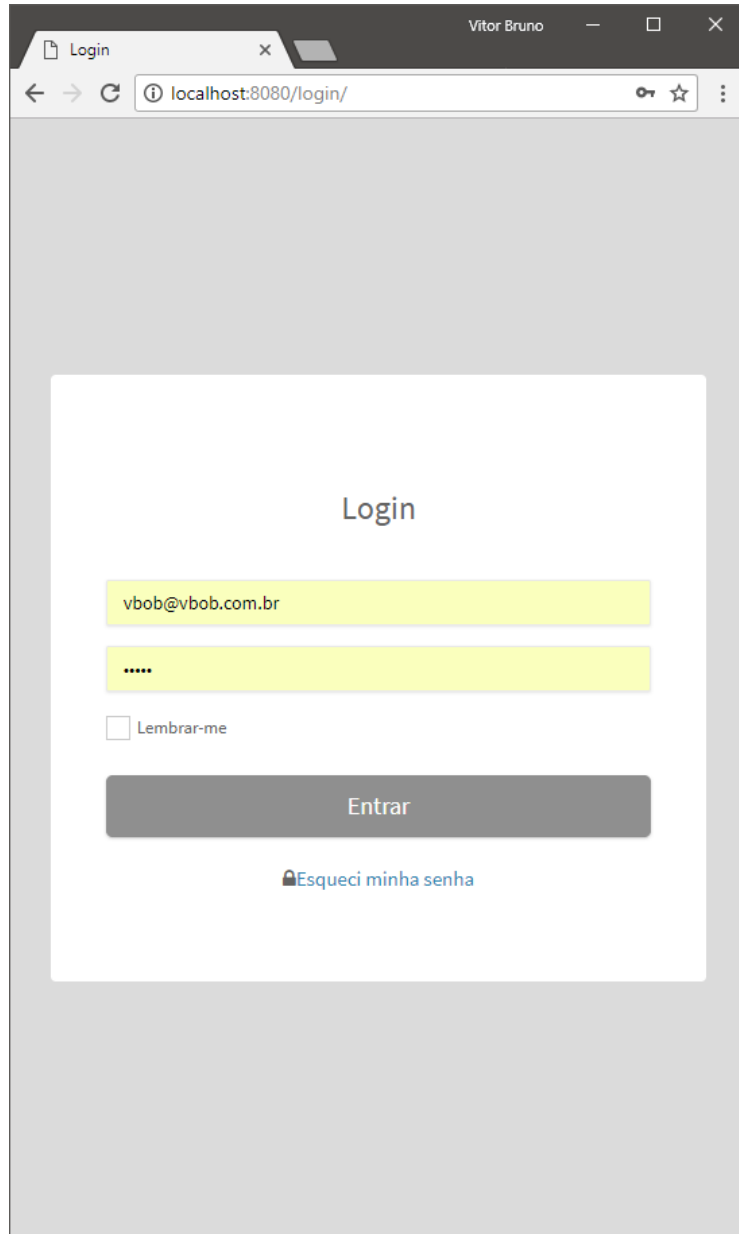
Entrar

[Esqueci minha senha](#)

Sistema para Escolas de Música

Versão: Alpha 0.0.1
Licenciado para: Desenvolvedor

Tela de Login (Celular): A tela de Login funciona perfeitamente no celular.



The image shows a mobile browser window with a dark header bar. The browser's address bar displays 'localhost:8080/login/'. The main content area has a light gray background. Centered on the screen is a white rectangular box containing the login form. The form is titled 'Login' in a dark gray font. Below the title are two yellow input fields: the first contains the email 'vbob@vbob.com.br' and the second contains masked characters '*****'. Under the password field is a checkbox labeled 'Lembrar-me'. Below these is a dark gray button with the text 'Entrar'. At the bottom of the form is a link that reads 'Esqueci minha senha' with a small lock icon to its left.

Login

vbob@vbob.com.br

☐ Lembrar-me

Entrar

[Esqueci minha senha](#)

Senha Inválida

A screenshot of a web browser window titled "Login" with the address bar showing "localhost:8080/login/". The page displays a "Login" form with the title "Login" and the message "Senha Incorreta" (Invalid Password). The email field contains "vbob@vbob.com.br" and the password field is masked with "*****". There is a "Lembrar-me" checkbox and an "Entrar" button. A link "Esqueci minha senha" is at the bottom.

Login

Senha Incorreta

vbob@vbob.com.br

☐ Lembrar-me

Entrar

[Esqueci minha senha](#)

Usuário Não Encontrado

A screenshot of a web browser window titled "Login" with the address bar showing "localhost:8080/login/". The page displays a "Login" form with the title "Login" and the message "Usuário não encontrado" (User not found). The email field contains "vbob@teste.com.br" and the password field is masked with "*****". There is a "Lembrar-me" checkbox and an "Entrar" button. A link "Esqueci minha senha" is at the bottom.

Login

Usuário não encontrado

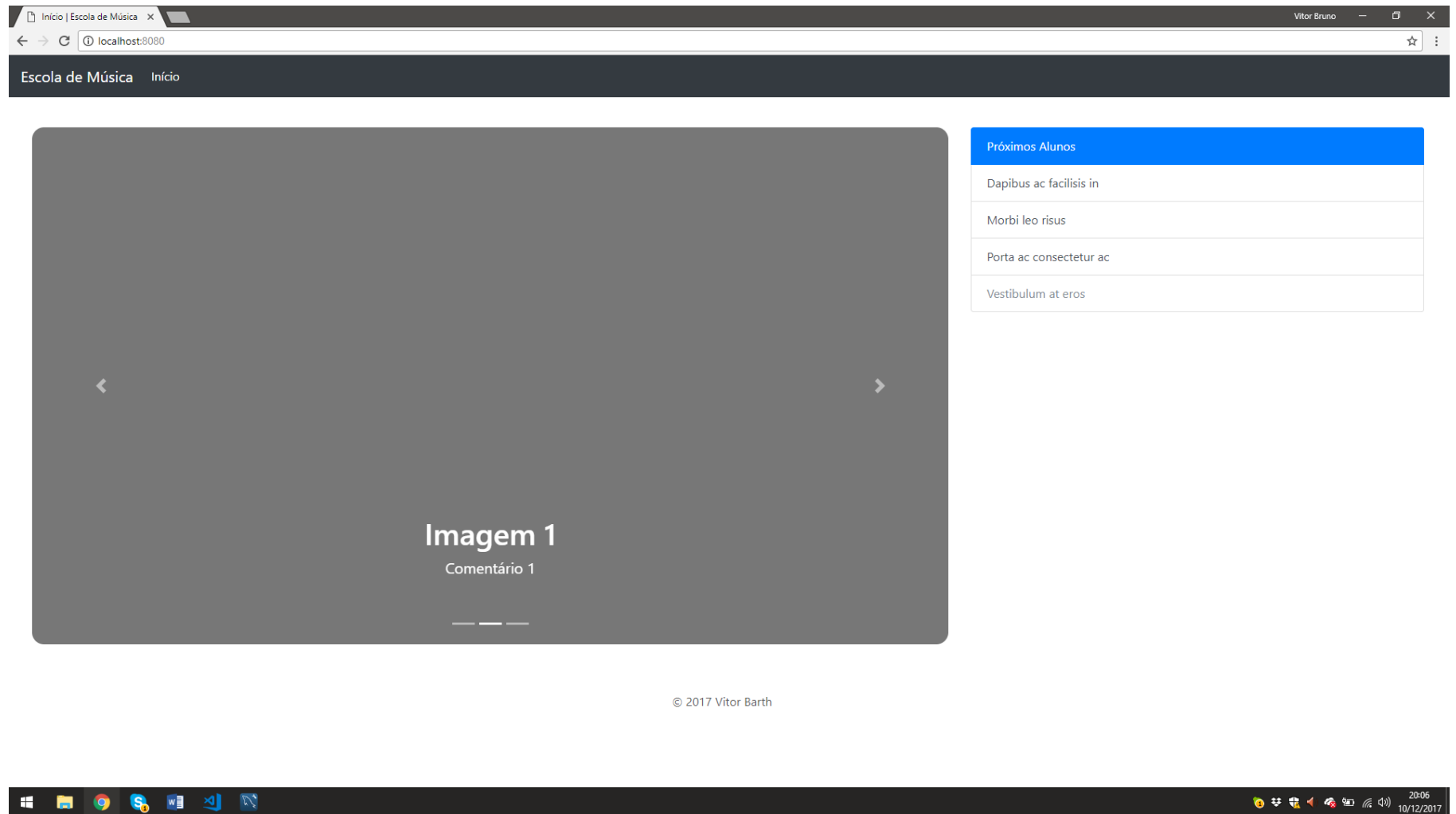
vbob@teste.com.br

☐ Lembrar-me

Entrar

[Esqueci minha senha](#)

Tela Principal: A tela principal está com todos os objetos previstos desenhados, faltando alguns ajustes para ficar mais agradável visualmente. Algumas das funcionalidades exigem trabalhos futuros, como cadastro de horários de aula.



Classe “Professor”

```
Professor.ts - escola_musica - Visual Studio Code
File Edit Selection View Go Debug Tasks Help

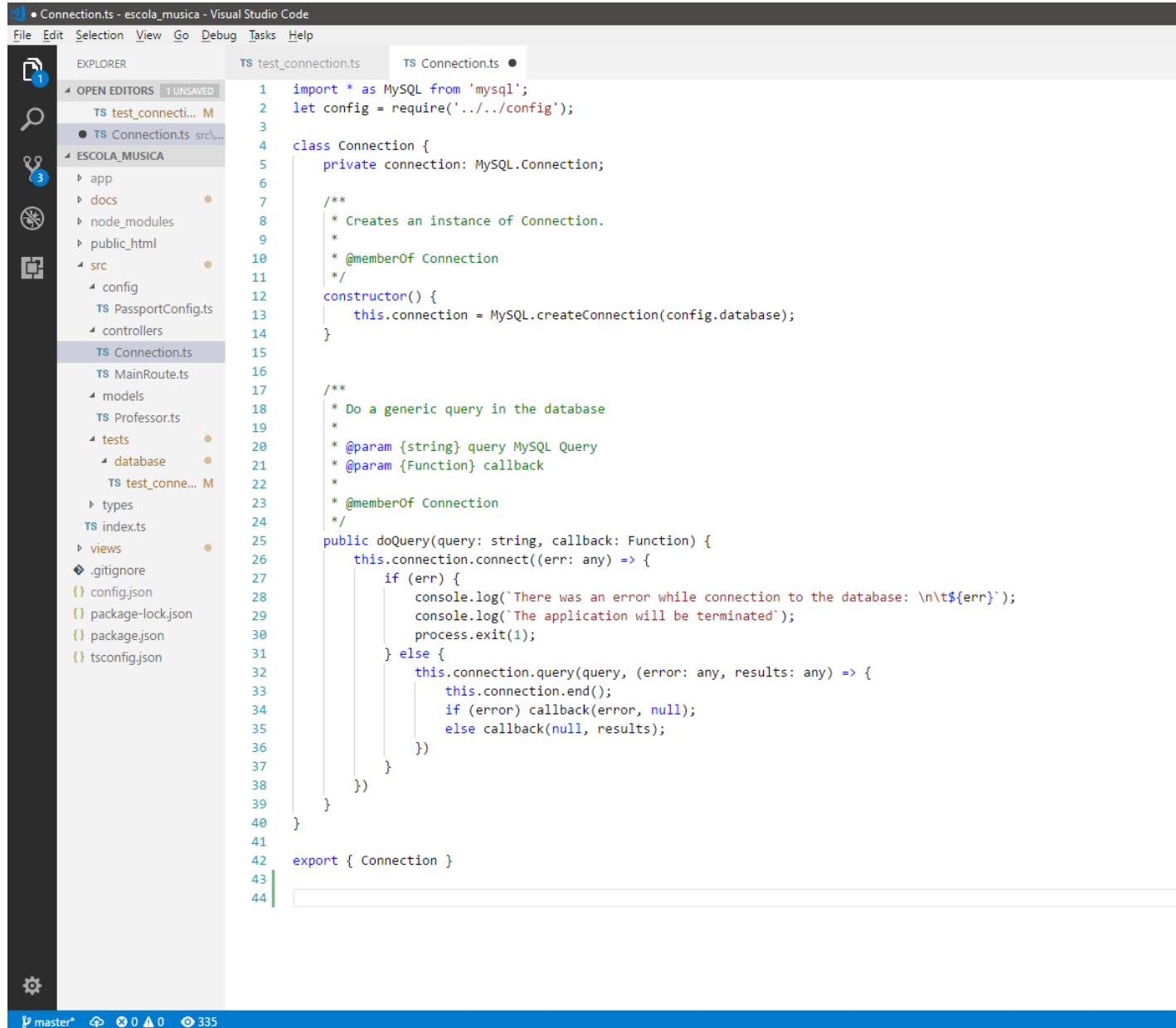
TS Professor.ts x
12
13 class Professor {
14   constructor(public nome: string, public telefone: string, public endereco: string,
15     public valor_hora_aula: Number, public email: string, public senha: string, public id_professor?: Number) {
16
17     if (id_professor)
18       this.id_professor = id_professor;
19
20     this.nome = nome;
21     this.telefone = telefone;
22     this.endereco = endereco;
23     this.valor_hora_aula = valor_hora_aula;
24     this.email = email;
25     this.senha = senha;
26   }
27
28   public static searchByEmail(email: string, callback:Function) {
29     let C: Connection = new Connection();
30     //console.log(`SELECT * FROM professor WHERE email = ${email}`)
31     C.doQuery(`SELECT * FROM professor WHERE email = '${email}'`, (error: any, results:Array<Professor>) => {
32       if (error) callback(error, null)
33       else callback(null, results[0])
34     })
35   }
36
37   public static searchById(id: Number, callback:Function) {
38     let C: Connection = new Connection();
39     //console.log(`SELECT * FROM professor WHERE id = ${id}`)
40     C.doQuery(`SELECT * FROM professor WHERE id_professor = ${id}`, (error: any, results:Array<Professor>) => {
41       if (error) callback(error, null)
42       else callback(null, results)
43     })
44   }
45
46   public static verifyPassword(hash: string, provided_password:string, callback: Function) {
47     BCrypt.compare(provided_password, hash, function(err, res) {
48       callback(res)
49     });
50   }
51 }
52
53 export { Professor }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Server started succesfully. Listening to port 8080

master 0 0 0 335

Conexão com o Banco de Dados



```
1 import * as MySQL from 'mysql';
2 let config = require('../../config');
3
4 class Connection {
5     private connection: MySQL.Connection;
6
7     /**
8      * Creates an instance of Connection.
9      *
10     * @memberOf Connection
11     */
12     constructor() {
13         this.connection = MySQL.createConnection(config.database);
14     }
15
16     /**
17     * Do a generic query in the database
18     *
19     * @param {string} query MySQL Query
20     * @param {Function} callback
21     *
22     * @memberOf Connection
23     */
24     public doQuery(query: string, callback: Function) {
25         this.connection.connect((err: any) => {
26             if (err) {
27                 console.log(`There was an error while connection to the database: \n\t${err}`);
28                 console.log(`The application will be terminated`);
29                 process.exit(1);
30             } else {
31                 this.connection.query(query, (error: any, results: any) => {
32                     this.connection.end();
33                     if (error) callback(error, null);
34                     else callback(null, results);
35                 })
36             }
37         })
38     }
39 }
40
41 export { Connection }
```

Testes de Conexão: Estão sendo realizados testes unitários das funcionalidades de classes de negócio.

The screenshot displays the Visual Studio Code interface. The Explorer sidebar on the left shows a project structure with folders like 'app', 'docs', 'node_modules', 'public_html', 'src', 'config', 'controllers', 'models', 'tests', and 'database'. The 'tests' folder is expanded, showing 'TS test_connection.ts' as the active file.

The main editor area shows the content of 'TS test_connection.ts':

```
1 import { Connection } from "../../controllers/Connection";
2 import { Professor } from "../../models/Professor"
3 import { expect } from 'chai'
4
5 describe("Connection", () => {
6     it("Acquire Connection", () => {
7         let c: Connection = new Connection();
8         c.doQuery("SELECT 1 + 1 AS solution", (error:any, results:any) => {
9             expect(error ? error : results[0].solution).to.equal(2);
10        })
11    })
12 })
13
14 describe("Professor", () => {
15     it("Search Professor by E-Mail", () => {
16         Professor.searchByEmail('vbob@vbob.com.br', (error: any, results:Array<Professor>) => {
17             expect(error ? error : results[0].email).to.equal('vbob@vbob.com.br');
18         })
19     })
20
21     it("Search Professor by Id", () => {
22         Professor.searchById(1, (error: any, results:Array<Professor>) => {
23             expect(error ? error : results[0].email).to.equal('vbob@vbob.com.br');
24         })
25     })
26 })
27
28
```

At the bottom, the 'TERMINAL' pane shows the command used to run the tests and the resulting output:

```
PS C:\Users\vbob\Documents\escola_musica\src> mocha -r ts-node/register ./tests/database/test_connection.ts

Connection
  ✓ Acquire Connection (79ms)

Professor
  ✓ Search Professor by E-Mail
  ✓ Search Professor by Id

3 passing (145ms)

PS C:\Users\vbob\Documents\escola_musica\src>
```

The status bar at the very bottom indicates the current branch is 'master' and shows 0 errors, 0 warnings, and 335 lines of code.