

SENAC

**TECNOLOGIA EDUCACIONAL: POTENCIALIZANDO EXPERIÊNCIA DO  
USUÁRIO ATRAVÉS DE UM AMBIENTE VIRTUAL DE APRENDIZAGEM  
ORGANIZADO E EFICIENTE**

AYRA WANDERLEY HIDASI  
GABRIELA GUEDES DA CRUZ  
RAFAEL DE LIMA SANTOS  
VITOR BORDIN GOMES

AYRA WANDERLEY HIDASI  
GABRIELA GUEDES DA CRUZ  
RAFAEL DE LIMA SANTOS  
VITOR BORDIN GOMES

**Tecnologia educacional:** potencializando experiência do usuário através de um ambiente virtual de aprendizagem organizado e eficiente

Projeto Integrador: implantação de Banco de Dados apresentado ao Senac, como exigência parcial para obtenção de nota no curso de Tecnologia em Banco de Dados.

## AGRADECIMENTOS

Aos colegas, alunos do curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Bianca Silva Barcelos e Kleverton Machado Kulmann que idealizaram e contribuíram na construção da primeira parte deste projeto.

## RESUMO

Uma plataforma virtual de ensino que tem como objetivo ser visualmente atrativa e descomplicada, com foco na experiência de usuário, tornar o fluxo de rotinas do aluno dentro da plataforma mais agradável e reduzir o número de desistências, pois a plataforma não sobrecarregará o aluno com informações que não são referentes às suas matérias daquele semestre e atividades respectivas. O modo com que a plataforma irá conduzir o aluno no primeiro acesso será como um tutorial inicial em sessões de foco como um passo a passo, e caso ele deseje poderá ter acesso a um vídeo que mostra cada passo com audiodescrição, desse modo poderemos abranger a funcionalidade da plataforma para mais pessoas, não importando se possuem ou não deficiências. Para o acesso dos professores na plataforma, um painel especializado e dedicado às necessidades que os mesmos terão para inserir conteúdo novo na plataforma, avaliar os alunos nas atividades e realizar as correções de avaliações de conteúdo do período, seja ele bimestre, trimestre ou semestre, um jeito usável e rápido desse modo o profissional poderá ter mais tempo para organizar suas rotinas na plataforma.

Palavras-chave: descomplicado, plataforma, acessibilidade, necessidades.

## **Sumário**

Sumário	5
1. Introdução	6
2. Implantação do Banco de Dados	7
2.1 Diagrama Entidade Relacionamento (DER)	7
2.2 Levantamento dos grupos de dados e afinidades entre eles	8
2.2.1 Identificação de grupos de dados	8
2.2.2 Identificação de afinidades entre os grupos de dados	9
2.2.3 Identificação de relacionamentos entre as entidades	10
3. Tabelas SQL Normalizadas na 3FN:	11
4. Modelo Físico Detalhado	12
4.1. COMANDOS DML	19
4.2. COMANDOS DDL	20
6. Referências	22

## **1. Introdução**

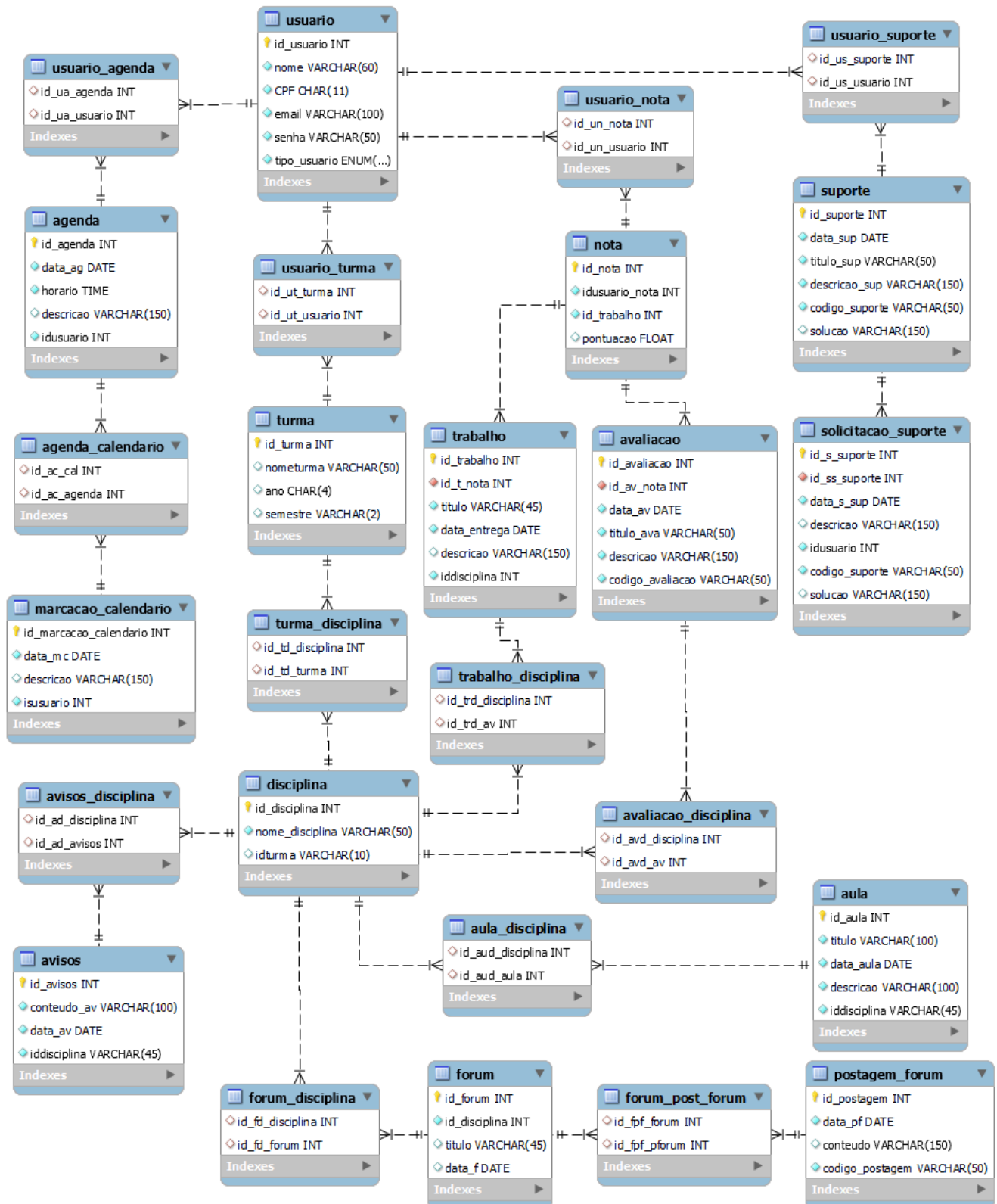
O produto proposto é um ambiente virtual de aprendizagem, voltado à melhor experiência do usuário ajudando a se organizar em suas aulas, atividades e avaliações de forma eficiente. Essa plataforma tem uma interface inicial (*dashboard*) limpa e organizada, que permite visualização rápida das informações essenciais para o estudante, como: matérias do semestre atual, notas separadas por sessões de cada matéria, acesso rápido ao fórum, resumo breve das entregas de atividades e avaliações, avisos da instituição e com versão mobile para o aluno ter liberdade de utilizar o sistema onde quiser.

A visão detalhada do produto, bem como o estudo da sua viabilidade e a especificação de seus requisitos foram tratados na primeira parte deste projeto juntamente com alunos de graduação de outros cursos Superiores de Tecnologia do Senac.

O desenvolvimento desse projeto foca na transformação do Diagrama Entidade Relacionamento (DER) referente ao projeto de ambiente virtual de aprendizagem em tabelas SQL, normalizadas e na terceira forma normal.

## 2. Implantação do Banco de Dados

### 2.1 Diagrama Entidade Relacionamento (DER)



## 2.2 Levantamento dos grupos de dados e afinidades entre eles

Abaixo estão a identificação dos grupos de dados e a afinidade entre eles.

### 2.2.1 Identificação de grupos de dados

Por se tratar de um software de ambiente virtual de aprendizagem, os grupos de dados identificados são: usuário, turma, disciplina, aula, trabalho, avaliação, nota, fórum, postagem no fórum, agenda, marcação no calendário, avisos, suporte e solicitação de suporte.

- **Usuário:** Representa os usuários do sistema, como alunos, professores, funcionários e clientes.
- **Turma:** Representa as turmas de cursos, com disciplina e código da turma.
- **Disciplina:** Representa as disciplinas dos cursos, com nome, código da disciplina e professor.
- **Aula:** Representa as aulas das disciplinas, com data, descrição e código da aula.
- **Trabalho:** Representa os trabalhos dos alunos, com data de entrega, descrição, código do trabalho e nota.
- **Avaliação:** Representa as avaliações das disciplinas pelos alunos, com data, descrição e código da avaliação.
- **Nota:** Representa as notas dos alunos nas disciplinas, com código da nota, código da turma, código da disciplina e código do trabalho.
- **Fórum:** Representa os fóruns de discussão das disciplinas, com nome, descrição e código do fórum.
- **Postagem do Fórum:** Representa as postagens dos usuários nos fóruns, com data, descrição e código da postagem.
- **Agenda:** Representa as agendas de agendamentos, incluindo a data, horário, descrição e data.



- **Marcação de Calendário:** Representa as marcações de calendário, com data, descrição e data.
- **Avisos:** Representa os avisos dos professores para os alunos
- **Suporte:** Representa os tickets de suporte técnico, com data, descrição, data, código do suporte e solução.

### 2.2.2 Identificação de afinidades entre os grupos de dados

Os grupos de dados manipularão as informações (atributos) como segue:

- **Usuário:** ID (chave primária), nome, CPF, e-mail, senha, tipo de usuário (aluno, professor, gestor)
- **Turma:** ID (chave primária), nome da turma, ano, semestre
- **Disciplina:** ID (chave primária), nome da disciplina, ID da turma (chave estrangeira)
- **Aula:** ID (chave primária), título, descrição, data, ID da disciplina (chave estrangeira)
- **Trabalho:** ID (chave primária), título, descrição, data de entrega, ID da disciplina (chave estrangeira)
- **Avaliação:** ID (chave primária), título, descrição, data de avaliação, ID da disciplina (chave estrangeira)
- **Nota:** ID (chave primária), ID do aluno (Chave Estrangeira), ID do trabalho/avaliação (chave estrangeira), valor
- **Fórum:** ID (chave primária), título, descrição, data, ID da disciplina (chave estrangeira)
- **Postagem no fórum:** ID (chave primária), conteúdo, data, ID do autor (chave estrangeira referenciando usuário), ID do fórum (chave estrangeira)
- **Agenda:** ID (chave primária), data, descrição, ID do usuário (chave estrangeira)
- **Marcação no Calendário:** ID (chave primária), data, descrição, ID do usuário (chave estrangeira)
- **Avisos:** ID (chave primária), conteúdo, data, ID da Disciplina (chave estrangeira)

- **Suporte:** ID (chave primária), título, descrição, data, ID do Usuário (chave estrangeira)
- **Solicitação de Suporte:** ID (chave primária), título, descrição, data, ID do usuário (chave estrangeira), ID do suporte (chave estrangeira)

### 2.2.3 Identificação de relacionamentos entre as entidades

Os relacionamentos entre as entidades no banco de dados do ambiente virtual de aprendizagem são os seguintes:

- **Usuário - Turma:** um usuário pode estar associado a uma ou mais turmas.
- **Turma - Disciplina:** uma turma pode ter várias disciplinas associadas.
- **Disciplina - Aula:** uma disciplina pode ter várias aulas.
- **Disciplina - Trabalho:** uma disciplina pode ter vários trabalhos associados.
- **Disciplina - Avaliação:** uma disciplina pode ter várias avaliações associadas.
- **Usuário - Nota:** um usuário (aluno) pode ter várias notas associadas.
- **Disciplina - Fórum:** uma disciplina pode ter vários fóruns associados.
- **Fórum - Postagem no Fórum:** um fórum pode ter várias postagens.
- **Usuário - Agenda:** um usuário (aluno ou professor) pode ter várias entradas na agenda.
- **Disciplina - Avisos:** uma disciplina pode ter vários avisos.
- **Usuário - Suporte:** um usuário (aluno ou professor) pode ter várias solicitações de suporte.
- **Suporte - Solicitação de Suporte:** um suporte deve ter apenas uma solicitação de suporte associada.

### 3. Tabelas SQL Normalizadas na 3FN:

A normalização de tabelas em um banco de dados é um processo fundamental para garantir a organização, confiabilidade e eficiência do armazenamento de informações. Através da normalização, minimizamos a redundância de dados, eliminamos anomalias e otimizamos o desempenho das consultas.

No DER original, a tabela “Agenda” incluía data de agendamento em um único atributo. Na estrutura SQL, os valores de “data” e “horário” foram separados em dois atributos distintos: Data\_Ag e Horario. Esta separação é uma prática que melhora a precisão dos dados, evitando redundâncias e facilitando consultas específicas por data ou horário. Seguindo a 1FN, que exige que os dados sejam armazenados em tabelas sem atributos multivalorados, essa mudança assegura que cada campo contenha apenas um valor atômico.

O CPF, presente no DER original, foi omitido na tabela “Usuario” no SQL. Essa decisão se deve a considerações de privacidade e segurança de dados, reduzindo o risco de exposição de informações sensíveis. A remoção de dados pessoais sensíveis diretamente do banco de dados está alinhada com as boas práticas de segurança e proteção de dados, atendendo aos requisitos da Lei Geral de Proteção de Dados (LGPD) no Brasil.

A tabela “Nota” no DER apresentava atributos que poderiam levar a dependências transitivas. Na versão SQL, a estrutura foi ajustada para incluir referências mais claras, com os seguintes campos: ID\_Nota, Codigo\_Nota, ID\_Turma, ID\_Disciplina e ID\_Trabalho. Essa reorganização garante que cada atributo dependa diretamente da chave primária, eliminando dependências transitivas e anomalias de inserção, atualização e exclusão, conforme a 3FN.

No DER, “Marcacao” no “Calendario” incluía atributos diretamente relacionados ao usuário. Na tabela SQL, um ID\_Agenda foi adicionado como chave estrangeira. Essa alteração assegura a integridade referencial, conectando cada marcação de calendário a uma entrada específica na agenda. A inclusão de chaves estrangeiras é fundamental para manter a consistência dos dados e garantir que todas as referências entre tabelas estejam corretamente estabelecidas.

A tabela Suporte no SQL foi expandida para incluir detalhes como Codigo\_Suporte e Solucao, além dos atributos originais ID\_Suporte, ID\_Usuario, Data, Descricao, e Destique. Essa modificação permite um rastreamento mais detalhado dos tickets de suporte e suas respectivas soluções, melhorando a gestão e o histórico de atendimentos. A manutenção de um registro detalhado e preciso dos suportes garante a disponibilidade de informações completas para análises futuras e auditorias.

#### **4. Modelo Físico Detalhado**

```
create table usuario (  
    id_usuario int primary key auto_increment,  
    nome varchar(60) not null,  
    CPF char (11) not null unique,  
    email varchar(100) not null unique,  
    senha varchar(50) not null unique,  
    tipo_usuario enum('aluno','professor','funcionario','cliente') not null  
);
```

```
create table agenda (  
    id_agenda int primary key auto_increment,  
    data_ag date not null,  
    horario time not null,  
    descricao varchar(150),  
    idusuario int not null  
);
```

```
create table marcacao_calendario (  
    id_marcacao_calendario int primary key auto_increment,  
    data_mc date not null,  
    descricao varchar(150),  
    isusuario int not null  
);
```

```
create table turma (  
    id_turma int primary key auto_increment,  
    nometurma varchar(50),  
    ano char(4),  
    semestre varchar(2)  
);
```

```
create table disciplina (  
    id_disciplina int primary key auto_increment,  
    nome_disciplina varchar(50) not null,  
    idturma varchar(10)  
);
```

```
create table avisos (  
    id_avisos int primary key auto_increment,  
    conteudo_av varchar(100) not null,  
    data_av date not null,  
    iddisciplina varchar(45) not null  
);
```

```
create table aula (  
    id_aula int primary key auto_increment,  
    titulo varchar(100) not null,  
    data_aula date not null,  
    descricao varchar(100) not null,  
    iddisciplina varchar(45) not null  
);
```

```
create table trabalho (  
    id_trabalho int primary key auto_increment,  
    id_t_nota int not null,  
    titulo varchar(45) not null,  
    data_entrega date not null,  
    descricao varchar(150),  
    iddisciplina int not null,  
    constraint fk_tn_nota foreign key (id_t_nota) references nota(id_nota)  
);
```

```
create table forum (  
    id_forum int primary key auto_increment,  
    id_disciplina int not null,  
    titulo varchar(45),  
    data_f date  
);
```

```
create table postagem_forum (  
    id_postagem int primary key auto_increment,  
    data_pf date not null,  
    conteudo varchar(150),  
    codigo_postagem varchar(50) not null  
);
```

```
create table suporte (  
    id_suporte int primary key auto_increment,  
    data_sup date not null,  
    titulo_sup varchar(50) not null,  
    descricao_sup varchar(150) not null,  
    codigo_suporte varchar(50) not null,  
    solucao varchar(150)  
);
```

```
create table solicitacao_suporte (  
    id_s_suporte int primary key auto_increment,  
    id_ss_suporte int not null,  
    data_s_sup date not null,  
    descricao varchar(150),  
    idusuario int not null,  
    codigo_suporte varchar(50) not null,  
    solucao varchar(150),  
    constraint fk_ss_suporte foreign key (id_ss_suporte) references suporte(id_suporte)  
);
```

```
create table nota (  
    id_nota int primary key auto_increment,  
    idusuario_nota int not null,  
    id_trabalho int not null,  
    pontuacao float  
);
```

```
create table avaliacao (  
    id_avaliacao int primary key auto_increment,  
    id_av_nota int not null,  
    data_av date not null,  
    titulo_ava varchar(50) not null,  
    descricao varchar(150)not null,  
    codigo_avaliacao varchar(50) not null unique,  
    constraint fk_av_nota foreign key (id_av_nota) references nota(id_nota)  
);
```

```
create table usuario_agenda (  
    id_ua_agenda int,  
    id_ua_usuario int,  
    constraint fk_ua_agenda foreign key (id_ua_agenda) references agenda(id_agenda),  
    constraint fk_ua_usuario foreign key (id_ua_usuario) references usuario(id_usuario)  
);
```

```
create table usuario_suporte (  
    id_us_suporte int,  
    id_us_usuario int,  
    constraint fk_us_suporte foreign key (id_us_suporte) references suporte(id_suporte),  
    constraint fk_us_usuario foreign key (id_us_usuario) references usuario(id_usuario)  
);
```

```
create table usuario_turma (  
    id_ut_turma int,  
    id_ut_usuario int,  
    constraint fk_ut_turma foreign key (id_ut_turma) references turma(id_turma),  
    constraint fk_ut_usuario foreign key (id_ut_usuario) references usuario(id_usuario)  
);
```

```
create table usuario_nota (  
    id_un_nota int,  
    id_un_usuario int,  
    constraint fk_un_turma foreign key (id_un_nota) references nota(id_nota),  
    constraint fk_un_usuario foreign key (id_un_usuario) references usuario(id_usuario)  
);
```



```
create table agenda_calendario (  
    id_ac_cal int,  
    id_ac_agenda int,  
    constraint fk_ac_cal foreign key (id_ac_cal) references  
    marcacao_calendario(id_marcacao_calendario),  
    constraint fk_ac_agenda foreign key (id_ac_agenda) references agenda(id_agenda)  
);
```

```
create table turma_disciplina (  
    id_td_disciplina int,  
    id_td_turma int,  
    constraint fk_td_disciplina foreign key (id_td_disciplina) references disciplina(id_disciplina),  
    constraint fk_td_turma foreign key (id_td_turma) references turma(id_turma)  
);
```

```
create table avisos_disciplina (  
    id_ad_disciplina int,  
    id_ad_avisos int,  
    constraint fk_ad_disciplina foreign key (id_ad_disciplina) references disciplina(id_disciplina),  
    constraint fk_ad_avisos foreign key (id_ad_avisos) references avisos(id_avisos)  
);
```

```
create table forum_disciplina (  
    id_fd_disciplina int,  
    id_fd_forum int,  
    constraint fk_fd_disciplina foreign key (id_fd_disciplina) references disciplina(id_disciplina),  
    constraint fk_fd_forum foreign key (id_fd_forum) references forum(id_forum)  
);
```

```
create table aula_disciplina (  
    id_aud_disciplina int,  
    id_aud_aula int,  
    constraint fk_aud_disciplina foreign key (id_aud_disciplina) references  
    disciplina(id_disciplina),  
    constraint fk_aud_forum foreign key (id_aud_aula) references aula(id_aula)  
);
```

```
create table avaliacao_disciplina (  
    id_avd_disciplina int,  
    id_avd_av int,  
    constraint fk_avd_disciplina foreign key (id_avd_disciplina) references  
    disciplina(id_disciplina),  
    constraint fk_avd_forum foreign key (id_avd_av) references avaliacao(id_avaliacao)  
);
```

```
create table trabalho_disciplina (  
    id_trd_disciplina int,  
    id_trd_av int,  
    constraint fk_trd_disciplina foreign key (id_trd_disciplina) references  
    disciplina(id_disciplina),  
    constraint fk_trd_forum foreign key (id_trd_av) references trabalho(id_trabalho)  
);
```

```
create table forum_post_forum (  
    id_fpf_forum int,  
    id_fpf_pforum int,  
    constraint fk_fpf_forum foreign key (id_fpf_forum) references forum(id_forum),  
    constraint fk_fpf_pforum foreign key (id_fpf_pforum) references  
    postagem_forum(id_postagem)  
);
```

#### 4.1. COMANDOS DML

-- INSERT

```
insert into usuario(nome, CPF, email, senha, tipo_usuario)
```

```
    values("Pedro", 45452022068,"pedrosp@live.com.br", "Senac@123","aluno"),  
          ("Marcia", 60052022074,"marciasp@gmail.com.br", "Senac@marcia","aluno"),  
          ("João", 70450722400,"joao.edu@hotmail.com.br", "775522","professor"),  
          ("Daiane", 48460022200,"daiane.fun@live.com.br", "fun@123","funcionario"),  
          ("Olávio", 70052055068,"olavio.cl@yahoo.com.br", "cl@987","cliente");
```

```
insert into nota(idusuario_nota, id_trabalho, pontuacao)
```

```
    values(1,1,8.5),  
          (2,2,7.5),  
          (3,3,5.0),  
          (4,4,8.3);
```

-- UPDATE

```
update usuario set senha = "senac@777" where id_usuario = 1;
```

```
update usuario set idade = 20 where id_usuario = 1;
```

```
update nota set pontuacao = 7.0 where id_nota = 3;
```

```
update nota set pontuacao = 7.9 where id_nota = 1;
```

-- DELETE

```
delete from usuario where id_usuario = 2;
```

```
delete from nota where id_nota = 4;
```

-- SELECT

```
select * from usuario;
```

```
select * from nota;
```

## 4.2. COMANDOS DDL

-- COMANDO DESC ou DESCRIBE

desc usuario;

describe nota;

-- SHOW

show tables;

show databases;

-- ALTER TABLE

alter table usuario add column idade int;

alter table usuario add column altura float, add column peso float;

alter table usuario drop column peso;

-- USE

use bd\_sistema\_info;

-- DROP

SET foreign\_key\_checks = 0;

drop table nota;

drop table nota;

drop database bd\_sistema\_info;

### 4.3. CORREÇÃO DE ERROS

Ao rodar o script no SGBD MySQL, o seguinte erro foi apontado:

*/Error Code: 3730. Cannot drop table 'nota' referenced by a foreign key  
constraint 'fk\_tn\_nota' on table 'trabalho'”*

Este erro ocorre porque a tabela nota está sendo referenciada por uma chave estrangeira na tabela trabalho. Para solucionar o erro e permitir a exclusão da tabela “nota”, desta forma, foi preciso remover a restrição de chave estrangeira antes de excluir a tabela “nota” através do comando SET foreign\_key\_checks = 0.

## **5. Conclusão**

Conclui-se que através da plataforma exposta anteriormente, podemos atingir nosso principal objetivo que é a diminuição do número de desistência dos alunos com os recursos que a plataforma tem a oferecer. Também agilizará o trabalho dos professores com as ferramentas oferecidas pelo software deixando-os com uma maior disponibilidade.

## **6. Referências**

MICROSOFT LEARN. **Descrição das noções básicas de normalização do banco de dados.** Solução de Problemas. Disponível em: < <https://learn.microsoft.com/pt-br/office/troubleshoot/access/database-normalization-description>> Acesso em 01/06/2024.