

Sumário

1.INTRODUÇÃO.....	2
2.OBJETIVO TÉCNICO.....	2
3.JUSTIFICATIVA.....	2
4.METODOLOGIAS.....	2
4.1. Modelo Conceitual.....	3
4.2. Modelo Lógico.....	4
5.IMPLEMENTAÇÃO DO PROJETO.....	4
6. DICIONÁRIO DE DADOS.....	6
ANEXO.....	6

1. INTRODUÇÃO

A utilização do banco de dados no mundo moderno garante a agilidade nas transações, bem como garantir a confiabilidade nas informações. Assim sendo, este projeto foi idealizado pelo professor Ricardo Nogueira para a disciplina de Engenharia de Software II, com a finalidade de integrar os alunos em um projeto.

O projeto consiste em um cadastro de discentes e docentes, que podem ter a fluência linguística em 1(um) ou vários idiomas. Os discentes devem ser identificados por um código, nome, idade, cidade, uf, país e idioma (as). Os docentes devem ser identificados por um código, nome, idade, salário e idioma (as).

2. OBJETIVO TÉCNICO

O presente documento tem a finalidade de apresentar as diversas fases de desenvolvimento do banco de dados do projeto em questão.

3. JUSTIFICATIVA

A elaboração do banco de dados iniciou após o processo de Análise de Requisitos. A pauta abrange as etapas de construção de uma base de dados hipotéticas criada com a finalidade de verificar os conceitos aprendidos em sala de aula.

O banco de dados final foi resultado de um levantamento detalhado de informações sobre o hipotético sistema de cadastramento de idiomas e posterior.

A elaboração do projeto contemplou as seguintes:

1. Modelagem Conceitual (DER)
2. Modelagem Lógica

4. METODOLOGIAS

Foram observadas as principais funcionalidades e ações realizadas, para então, se determinar quais seriam as entidades e os seus atributos relevantes. Em seguida, foi determinada a relação entre essas entidades e os atributos dessas relações.

De posse desses elementos, foi efetuada uma normalização do mesmo de forma a evitar redundância e problemas semânticos no banco de dados. A partir dessa normalização, foram criados os dicionários de dados para cada um desses sistemas, separadamente.

Finalizando essa etapa, foi feita a modelagem dos sistemas, separadamente, utilizando-se o software Workbench. Foi utilizado o método semântico para a modelagem de banco de dados denominado MER – Modelo Entidade-Relacionamento.

4.1. Modelo Conceitual

A construção deste modelo conceitual tem a finalidade de mostrar ao cliente os principais aspectos do banco de dados, assim como permitir uma interação mínima do usuário final com a tecnologia de banco de dados. Dessa forma, é possível a compreensão desse usuário de modo a garantir correção e respeito às regras de negócio por ele impostas.

Podemos verificar na ilustração abaixo a representação do banco de dados através do DER (Diagrama Entidade Relacionamento).

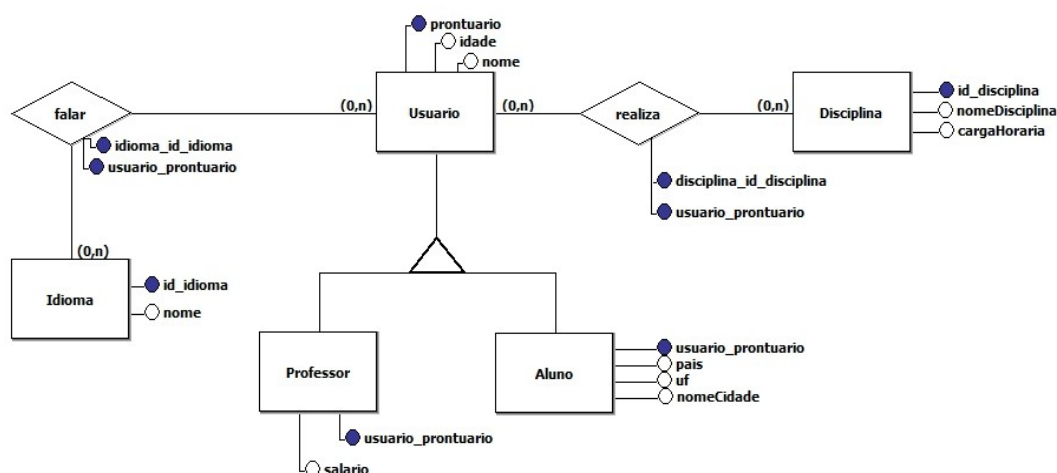


Figura 1 – Diagrama Entidade Relacionamento

4.2. Modelo Lógico

A partir de discussões e conclusões propostas, estruturou-se o modelo de dados considerando-se as necessidades de armazenamento de dados desejadas, de forma a garantir a extração de qualquer tipo de informação com segurança, correção e integridade das informações.

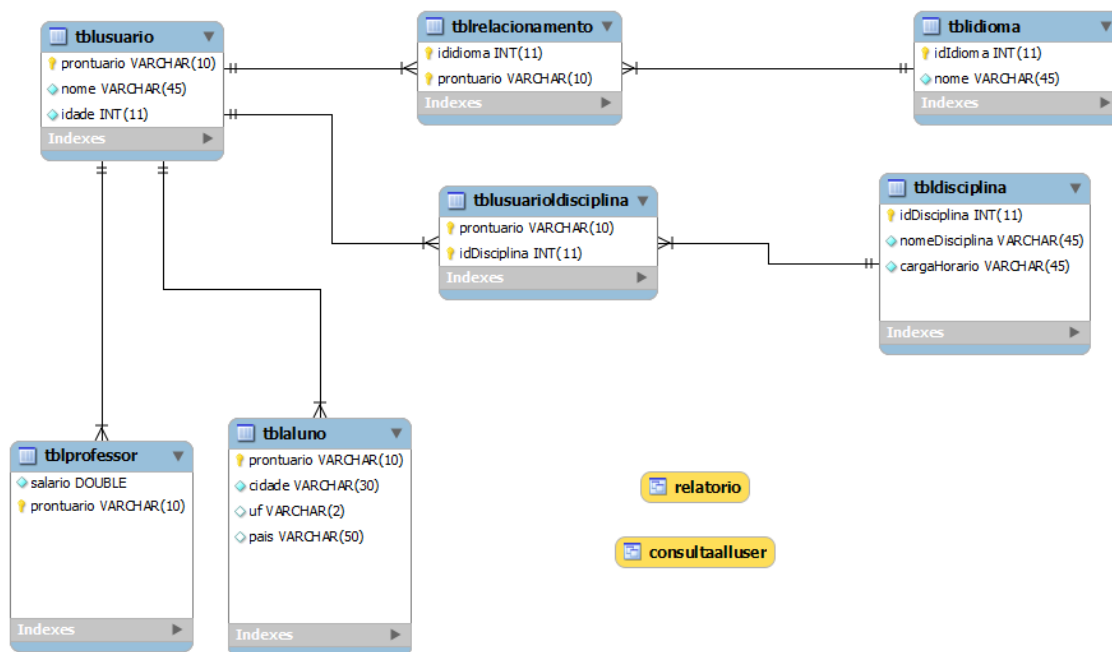


Figura 2 - Modelo Entidade Relacionamento

5. IMPLEMENTAÇÃO DO PROJETO

Com o modelo de banco de dados concluído, basta migrar para o gerenciador de banco de dados, o qual se optou por utilizar o MySQL.

O script foi gerado pela ferramenta Workbench. O esquema gerado pode ser visto em anexo.

6. DICIONÁRIO DE DADOS

tblUsuario

CAMPO	TIPO	NULO	EXTRAS
Prontuario	Varchar(10)	Não	Primary Key, Unique index
Nome	Varchar(45)	Não	
Idade	Int	Não	

tblProfessor

CAMPO	TIPO	NULO	EXTRAS
Prontuario	Varchar(10)	Não	Foreign Key
Salario	Double	Não	

tblAluno

CAMPO	TIPO	NULO	EXTRAS
Prontuario	Varchar(10)	Não	Foreign Key
Cidade	Varchar(30)	Não	
Uf	Varchar(2)	Sim	
País	Varchar(50)	Sim	

tblIdioma

CAMPO	TIPO	NULO	EXTRAS
idIdioma	Int	Não	Primary Key, Unique index, Auto Incremento
Nome	Varchar(45)	Não	Unique index

tblDisciplina

CAMPO	TIPO	NULO	EXTRAS
IdDisciplina	Int	Não	Primary Key, Unique Index, Auto Incremento
nomeDisciplina	Varchar(45)	Não	
cargaHoraria	Varchar(45)	Não	

tblRelacionamento

CAMPO	TIPO	NULO	EXTRAS
IdIdioma	Int	Não	Foreign Key
Prontuario	Varchar(10)	Não	Foreign Key, Index

tblUsuarioDisciplina

CAMPO	TIPO	NULO	EXTRAS
Prontuario	Varchar(10)	Não	Foreign Key, Index
IdDisciplina	Int	Não	Foreign Key, Index

ANEXO

Segue abaixo o script na linguagem SQL do banco de dados Projeto.

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';
```

```
CREATE SCHEMA IF NOT EXISTS `projeto` DEFAULT CHARACTER SET latin1 COLLATE
latin1_swedish_ci ;
USE `projeto`;
```

```
-----
-- Table `projeto`.`tblIdioma`
-----
```

```
CREATE TABLE IF NOT EXISTS `projeto`.`tblIdioma` (
  `idIdioma` INT NOT NULL AUTO_INCREMENT,
  `nome` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idIdioma`),
  UNIQUE INDEX `idIdioma_UNIQUE` (`idIdioma` ASC),
  UNIQUE INDEX `nome_UNIQUE` (`nome` ASC) )
ENGINE = InnoDB;
```

```
-----
-- Table `projeto`.`tblUsuario`
-----
```

```
CREATE TABLE IF NOT EXISTS `projeto`.`tblUsuario` (
  `prontuario` VARCHAR(10) NOT NULL,
  `nome` VARCHAR(45) NOT NULL,
  `idade` INT NOT NULL,
  PRIMARY KEY (`prontuario`),
  UNIQUE INDEX `prontuario_UNIQUE` (`prontuario` ASC) )
ENGINE = InnoDB;
```

```
-----
-- Table `projeto`.`tblAluno`
-----
```

```
CREATE TABLE IF NOT EXISTS `projeto`.`tblAluno` (
  `prontuario` VARCHAR(10) NOT NULL,
  `cidade` VARCHAR(30) NOT NULL,
  `uf` VARCHAR(2) NULL,
  `pais` VARCHAR(50) NULL,
  PRIMARY KEY (`prontuario`),
  INDEX `fk_Aluno_Usuario1` (`prontuario` ASC),
  CONSTRAINT `fk_Aluno_Usuario1`
    FOREIGN KEY (`prontuario`)
    REFERENCES `projeto`.`tblUsuario` (`prontuario`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-----
-- Table `projeto`.`tblProfessor`
-----
```

```
CREATE TABLE IF NOT EXISTS `projeto`.`tblProfessor` (
  `salario` DOUBLE NOT NULL,
  `prontuario` VARCHAR(10) NOT NULL,
  PRIMARY KEY (`prontuario`),
  CONSTRAINT `fk_Professor_Usuario1`
    FOREIGN KEY (`prontuario`)
    REFERENCES `projeto`.`tblUsuario` (`prontuario`)
```

```
REFERENCES `projeto`.`tblUsuario` (`prontuario` )  
ON DELETE NO ACTION  
ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-----  
-- Table `projeto`.`tblRelacionamento`  
-----
```

```
CREATE TABLE IF NOT EXISTS `projeto`.`tblRelacionamento` (  
  `idioma` INT NOT NULL ,  
  `prontuario` VARCHAR(10) NOT NULL ,  
  PRIMARY KEY (`idioma`, `prontuario`),  
  INDEX `fk_tblRelacionamento_Usuario1` (`prontuario` ASC),  
  CONSTRAINT `fk_tblRelacionamento_Idioma1`  
    FOREIGN KEY (`idioma` )  
    REFERENCES `projeto`.`tblIdioma` (`idIdioma` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_tblRelacionamento_Usuario1`  
    FOREIGN KEY (`prontuario` )  
    REFERENCES `projeto`.`tblUsuario` (`prontuario` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-----  
-- Table `projeto`.`tblDisciplina`  
-----
```

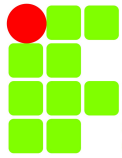
```
CREATE TABLE IF NOT EXISTS `projeto`.`tblDisciplina` (  
  `idDisciplina` INT NOT NULL AUTO_INCREMENT ,  
  `nomeDisciplina` VARCHAR(45) NOT NULL ,  
  `cargaHorario` VARCHAR(45) NOT NULL ,  
  PRIMARY KEY (`idDisciplina`),  
  UNIQUE INDEX `idDisciplina_UNIQUE` (`idDisciplina` ASC))  
ENGINE = InnoDB;
```

```
-----  
-- Table `projeto`.`tblUsuariolDisciplina`  
-----
```

```
CREATE TABLE IF NOT EXISTS `projeto`.`tblUsuariolDisciplina` (  
  `prontuario` VARCHAR(10) NOT NULL ,  
  `idDisciplina` INT NOT NULL ,  
  PRIMARY KEY (`prontuario`, `idDisciplina`),  
  INDEX `fk_tblUsuario_has_tblDisciplina_tblDisciplina1` (`idDisciplina` ASC),  
  INDEX `fk_tblUsuario_has_tblDisciplina_tblUsuario1` (`prontuario` ASC),  
  CONSTRAINT `fk_tblUsuario_has_tblDisciplina_tblUsuario1`  
    FOREIGN KEY (`prontuario` )  
    REFERENCES `projeto`.`tblUsuario` (`prontuario` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_tblUsuario_has_tblDisciplina_tblDisciplina1`  
    FOREIGN KEY (`idDisciplina` )  
    REFERENCES `projeto`.`tblDisciplina` (`idDisciplina` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
INSERT INTO `projeto`.`tblidioma` (`idIdioma`, `nome`) VALUES (1, 'INGLES');  
INSERT INTO `projeto`.`tblidioma` (`idIdioma`, `nome`) VALUES (2, 'PORTUQUES');  
INSERT INTO `projeto`.`tblidioma` (`idIdioma`, `nome`) VALUES (3, 'ESPANHOL');
```

```
INSERT INTO `projeto`.`tbldisciplina` (`idDisciplina`, `nomeDisciplina`, `cargaHorario`) VALUES (1,  
'INGLES', '200');  
INSERT INTO `projeto`.`tbldisciplina` (`idDisciplina`, `nomeDisciplina`, `cargaHorario`) VALUES (2,  
'PORTUQUES', '300');
```



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Capivari

Documentação do Projeto de Banco de Dados

Prof. Ricardo Nogueira de Figueiredo

```
INSERT INTO `projeto`.`tbldisciplina` (`idDisciplina`, `nomeDisciplina`, `cargaHorario`) VALUES (3, 'MATEMATICA', '400');
```

```
-- criar view Relatorio
drop view if exists relatorio;
create view relatorio as
SELECT u.nome, u.prontuario, i.nome as idioma
FROM tblusuario u
inner join tblrelacionamento r on (u.prontuario = r.prontuario)
inner join tblidioma i on (r.ididioma = i.idIdioma)
order by u.nome;
```

-- Criar View Consulta todos os Usuários

```
DROP VIEW IF EXISTS ConsultaAllUser;
CREATE VIEW ConsultaAllUser AS
SELECT u.nome, i.nome as idioma
FROM tblusuario AS u, tblidioma AS i
INNER JOIN tblrelacionamento AS r ON i.idIdioma= r.ididioma
WHERE u.prontuario = r.prontuario
ORDER BY u.nome ASC;
```

```
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```