UNIVERSIDADE FEDERAL DE SERGIPE

UNIMANAGE: GERENCIADOR DE PROJETOS

Especificações de Projeto, Modelo Conceitual, Modelo Lógico e DDL

Disciplina: Banco de Dados (COMP0455)

Turma: 03

Componentes da Equipe:

Carlos Henrique Santos Silveira

Guilherme Borges Cunha

Laura Cardoso Rodrigues

São Cristóvão/2023

1. INTRODUÇÃO

UniManage é um software de gestão de projetos que tem como objetivo principal auxiliar na organização de projetos acadêmicos supervisionados por um professor e executados por um time de estudantes. O sistema permitirá a criação, leitura, edição e exclusão de projetos, bem como a gestão de eventos relacionados a eles. O software possui dois tipos de usuário: o usuário Instructor, que é capaz de orientar projetos, agendar reuniões e eventos; E o usuário Student que será orientado no projeto. Ambos os usuários farão parte de uma determinada Instituição, e, no desenvolvimento de um mesmo projeto, farão parte de uma Equipe.

2. FUNCIONALIDADES

2.1. Gerenciamento de projetos

- Criação de projetos com informações sobre o nome, descrição e data de início e término previstos.
- Atribuição de um professor orientador ao projeto e seleção de um time de estudantes para trabalhar nele.
- Possibilidade de edição e exclusão de projetos.

2.2. Gerenciamento de eventos

- Criação de eventos relacionados ao projeto, como reuniões de equipe, apresentações e avaliações.
- Agendamento de datas e horários para cada evento e definição de participantes.
- Notificação automática dos participantes sobre os eventos agendados.

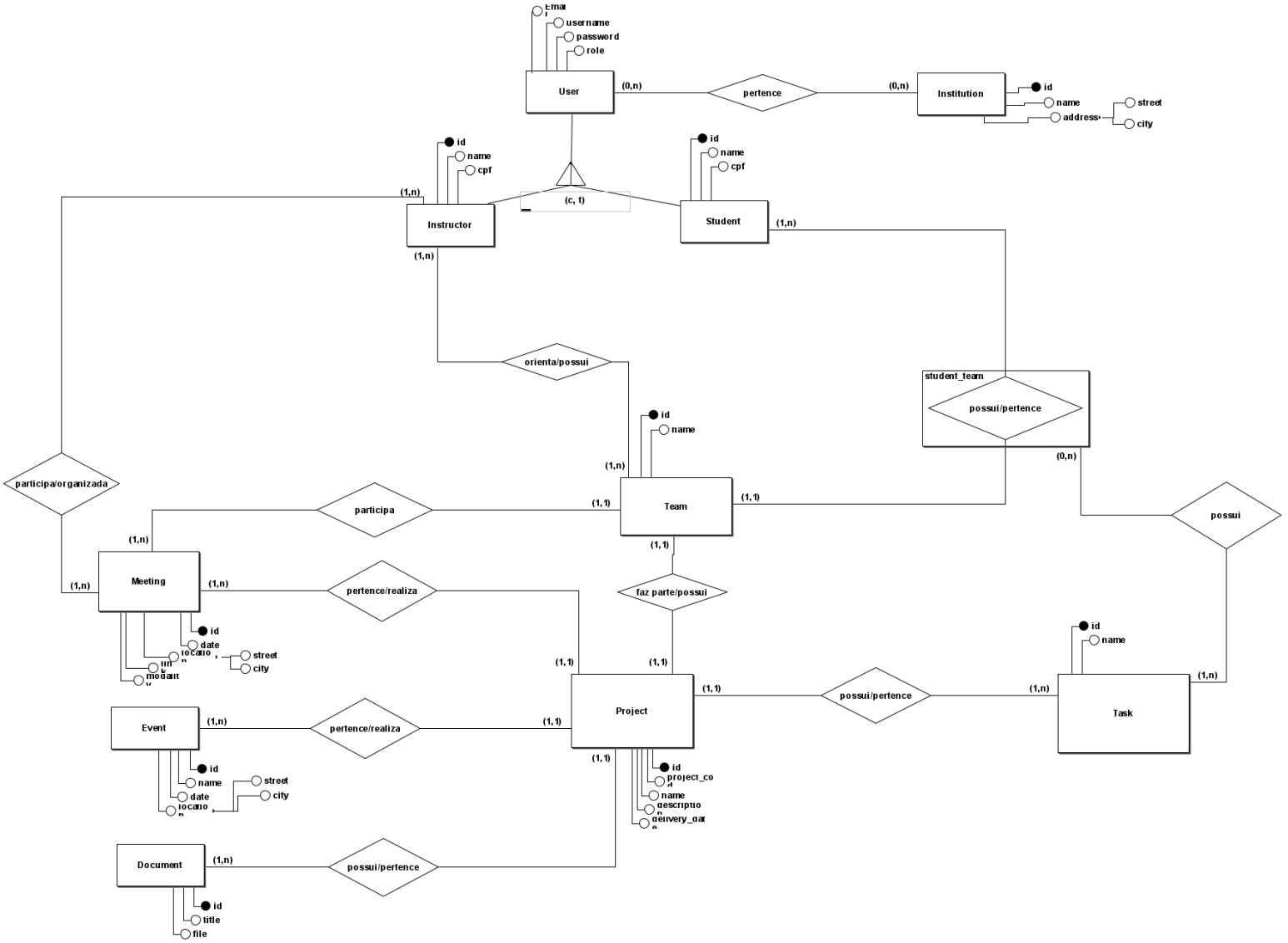
2.3. Controle de acesso e segurança

- Definição de perfis de usuário com diferentes níveis de acesso e permissões.
- Autenticação e autorização de usuários com base em login e senha.
- Armazenamento seguro de informações confidenciais do projeto e dos usuários.

3. TECNOLOGIAS UTILIZADAS

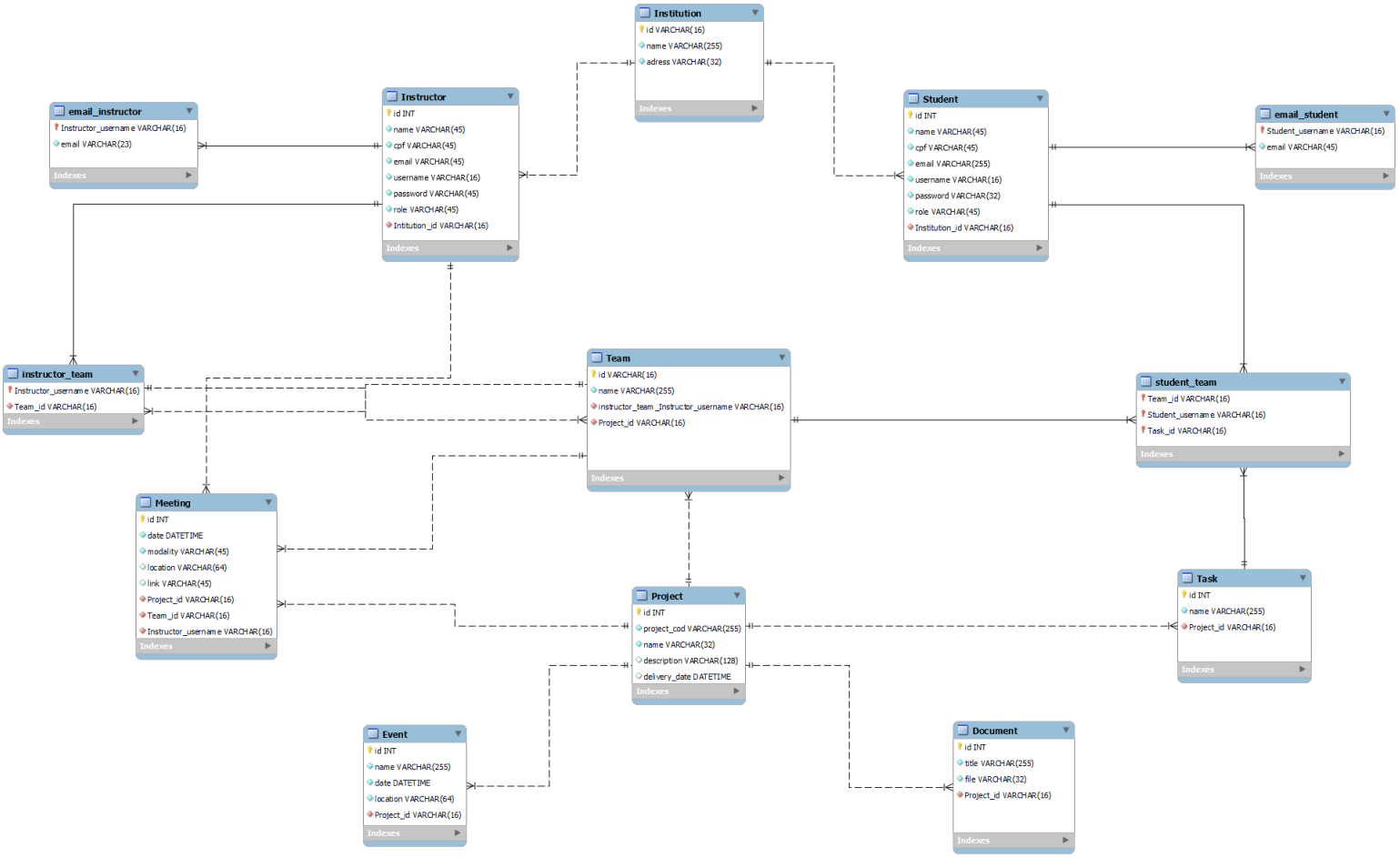
Ferramenta	Descrição
Sistema Operacional	Windows 10 ou superior
Servidor Web	Apache 2.4 ou Nginx 1.16
Banco de Dados	MySQL 5.7 ou superior
PHP	PHP 7.4 ou superior
Composer	Gerenciador de dependências do PHP
Laravel	Framework de aplicação WEB com arquitetura MVC em PHP
Bootstrap	Framework front-end para desenvolvimento de interfaces WEB
IDE	Visual Studio Code ou PHPStorm
Git	Sistema de controle de versão

4. MODELO CONCEITUAL



OBS: O modelo não possui entidade fraca devido ao fato de que todas as entidades possuem id próprio.

5. MODELO LÓGICO



6. SCRIPT DE CRIAÇÃO

Código (.sql):

-- AS COLUNAS DE UPDATED_AT E CREATED_AT SÓ FORAM USADAS PORQUE SÃO COLUNAS OBRIGATÓRIAS PARA FAZER REQUISIÇÕES COM O FRAMEWORK LARAVEL

CREATE DATABASE `demo_unimanage`;

USE `demo_unimanage`;

CREATE TABLE IF NOT EXISTS `user` (

`id` int(11) NOT NULL AUTO_INCREMENT,

`email` varchar(200) NOT NULL,

`username` varchar(100) NOT NULL,

`password` varchar(100) NOT NULL,

`role` varchar(20) NOT NULL,

`updated_at` timestamp NULL,

`created_at` timestamp NULL,

PRIMARY KEY (`id`)

```
);
CREATE TABLE IF NOT EXISTS `institution` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `name` varchar(200) NOT NULL,
    `address` varchar(200) NOT NULL,
    `updated_at` timestamp NULL,
    `created_at` timestamp NULL,
    PRIMARY KEY (`id`)
);
CREATE TABLE IF NOT EXISTS `instructor` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `name` varchar(200) NOT NULL,
    `cpf` varchar(20) NOT NULL,
    `user_fk` int(11) NOT NULL,
    `institution_fk` int(11) NOT NULL,
    `updated_at` timestamp NULL,
    `created_at` timestamp NULL,
    PRIMARY KEY (`id`),
    CONSTRAINT `user_fk_instructor` FOREIGN KEY (`user_fk`) REFERENCES `user` (`id`),
    CONSTRAINT `institution_fk_instructor` FOREIGN KEY (`institution_fk`) REFERENCES `institution`
(`id`)
);
CREATE TABLE IF NOT EXISTS `student` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `name` varchar(200) NOT NULL,
    `cpf` varchar(20) NOT NULL,
    `user_fk` int(11) NOT NULL,
    `institution_fk` int(11) NOT NULL,
    `updated_at` timestamp NULL,
    `created_at` timestamp NULL,
    PRIMARY KEY (`id`),
    CONSTRAINT `user_fk_student` FOREIGN KEY (`user_fk`) REFERENCES `user` (`id`),
    CONSTRAINT `institution_fk_student` FOREIGN KEY (`institution_fk`) REFERENCES `institution` (`id`)
);
CREATE TABLE IF NOT EXISTS `project` (
```

```
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `project_cod` varchar(20) NOT NULL,
    `name` varchar(200) NOT NULL,
    `description` varchar(200) NOT NULL,
    `delivery_date` date NOT NULL,
    `updated_at` timestamp NULL,
    `created_at` timestamp NULL,
    PRIMARY KEY (`id`)
);
CREATE TABLE IF NOT EXISTS `team` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `name` varchar(200) NOT NULL,
    `project_fk` int(11) NOT NULL,
    `updated_at` timestamp NULL,
    `created_at` timestamp NULL,
    PRIMARY KEY (`id`),
    CONSTRAINT `project_fk_team` FOREIGN KEY (`project_fk`) REFERENCES `project` (`id`)
);
CREATE TABLE IF NOT EXISTS `task` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `name` varchar(200) NOT NULL,
    `project_fk` int(11) NOT NULL,
    `updated_at` timestamp NULL,
    `created_at` timestamp NULL,
    PRIMARY KEY (`id`),
    CONSTRAINT `project_fk_task` FOREIGN KEY (`project_fk`) REFERENCES `project` (`id`)
);
CREATE TABLE IF NOT EXISTS `document` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `title` varchar(200) NOT NULL,
    `file` varchar(200) NOT NULL,
    `project_fk` int(11) NOT NULL,
    `updated_at` timestamp NULL,
    `created_at` timestamp NULL,
```

```
    PRIMARY KEY (`id`),
    CONSTRAINT `project_fk_document` FOREIGN KEY (`project_fk`) REFERENCES `project` (`id`)
);
CREATE TABLE IF NOT EXISTS `event` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `name` varchar(200) NOT NULL,
    `data` date NOT NULL,
    `location` varchar(200) NOT NULL,
    `project_fk` int(11) NOT NULL,
    `updated_at` timestamp NULL,
    `created_at` timestamp NULL,
    PRIMARY KEY (`id`),
    CONSTRAINT `project_fk_event` FOREIGN KEY (`project_fk`) REFERENCES `project` (`id`)
);

CREATE TABLE IF NOT EXISTS `meeting` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `date` date NOT NULL,
    `project_fk` int(11) NOT NULL,
    `location` varchar(200) NULL,
    `link` varchar(200) NULL,
    `modality` tinyint NOT NULL,
    `team_fk` int(11) NOT NULL,
    `instructor_fk` int(11) NOT NULL,
    `updated_at` timestamp NULL,
    `created_at` timestamp NULL,
    PRIMARY KEY (`id`),
    CONSTRAINT `project_fk_meeting` FOREIGN KEY (`project_fk`) REFERENCES `project` (`id`),
    CONSTRAINT `team_fk_meeting` FOREIGN KEY (`team_fk`) REFERENCES `team` (`id`),
    CONSTRAINT `instructor_fk_meeting` FOREIGN KEY (`instructor_fk`) REFERENCES `instructor` (`id`)
);
CREATE TABLE IF NOT EXISTS `student_team` (
    `team_fk` int(11) NOT NULL,
    `student_fk` int(11) NOT NULL,
```

```
  `updated_at` timestamp NULL,
  `created_at` timestamp NULL,
  CONSTRAINT `team_fk_student_team` FOREIGN KEY (`team_fk`) REFERENCES `team` (`id`),
  CONSTRAINT `student_fk_student_team` FOREIGN KEY (`student_fk`) REFERENCES `student` (`id`)
);
CREATE TABLE IF NOT EXISTS `instructor_team` (
  `team_fk` int(11) NOT NULL,
  `instructor_fk` int(11) NOT NULL,
  `updated_at` timestamp NULL,
  `created_at` timestamp NULL,
  CONSTRAINT `team_fk_instructor_team` FOREIGN KEY (`team_fk`) REFERENCES `team` (`id`),
  CONSTRAINT `instructor_fk_instructor_team` FOREIGN KEY (`instructor_fk`) REFERENCES `instructor` (`id`)
);
```

7. CONCLUSÃO

O software será desenvolvido seguindo as melhores práticas de desenvolvimento de software, incluindo metodologia ágil, testes automatizados, documentação de código e gestão de versão utilizando Git. O objetivo é fornecer um software robusto, seguro e fácil de usar para facilitar a gestão de projetos acadêmicos.

Com o “UniManage”, professores e estudantes terão uma ferramenta eficiente para coordenar e colaborar em projetos acadêmicos, garantindo a organização, comunicação e acompanhamento do andamento do trabalho.