

**SERVIÇO NACIONAL DE APRENDIZAGEM COMERCIAL**  
**SENAC**

**CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS**

**PROJETO INTEGRADOR III: DESENVOLVIMENTO DE  
SISTEMAS DINÂMICOS BASEADOS NA WEB**

Integrantes do grupo:

Carla Alecssandra Bruckmann

Gabriel Albuquerque Silva

João Victor da Silva Caldas

Larissa Lima Barreis

Mariana de Carvalho Fernandes

Nikolas Isler da Costa

Pedro Henrique de Oliveira Soares

Ryan Felipe Bernardo da Silva

Integrantes do grupo:

Carla Alecssandra Bruckmann

Gabriel Albuquerque Silva

João Victor da Silva Caldas

Larissa Lima Barreis

Mariana de Carvalho Fernandes

Nikolas Isler da Costa

Pedro Henrique de Oliveira Soares

Ryan Felipe Bernardo da Silva

## **PROJETO INTEGRADOR III: DESENVOLVIMENTO DE SISTEMAS DINÂMICOS BASEADOS NA WEB**

Professor Sr. Enoque Leal

### **TRABALHO PARA APROVAÇÃO EM DISCIPLINA**

## **Resumo**

Este trabalho introduz a modelagem de um sistema de gestão escolar usando UML. A primeira etapa do Projeto Integrador visa representar casos de uso, cenários de interação e a estrutura de classes. A modelagem detalhada facilita a compreensão e comunicação entre os membros da equipe. A elaboração de diagramas de casos de uso, cenários de interação e diagrama de classe abrangente captura as principais funcionalidades e entidades do sistema. Essa modelagem serve como base sólida para o desenvolvimento futuro, orientando a implementação de forma eficiente. Espera-se que essa etapa contribua para uma compreensão compartilhada dos requisitos e funcionalidades entre os membros da equipe.

## Sumário

1. Introdução.....	5
2. Diagrama de Casos .....	6
3. Descrição de Cenários.....	8
4. Diagrama de Classe .....	10
5. Dump MySQL .....	11
Conclusão .....	19
Referências .....	20

# 1. Introdução

A modelagem de software desempenha um papel crucial no desenvolvimento de sistemas de informação, fornecendo uma representação visual dos requisitos, estrutura e comportamento do sistema. Nesse contexto, a Linguagem Unificada de Modelagem (UML) se destaca como uma ferramenta poderosa para os desenvolvedores de software, permitindo a criação de diagramas padronizados que facilitam a compreensão e a comunicação entre os membros da equipe.

Este trabalho apresenta a primeira entrega do Projeto Integrador, focada na modelagem de um sistema de gestão escolar para um centro universitário fictício. O objetivo principal é utilizar os conceitos e técnicas da UML para representar os casos de uso, cenários de interação e a estrutura de classes do sistema proposto.

Ao longo deste documento, serão desenvolvidos diagramas de casos de uso que abrangem as principais funcionalidades do sistema, juntamente com descrições detalhadas dos cenários de interação. Além disso, será elaborado um diagrama de classe que reflete a estrutura de dados necessária para suportar as operações do sistema, incluindo entidades como alunos, professores, fornecedores e instituições jurídicas.

Esta etapa inicial do Projeto Integrador é fundamental para estabelecer uma base sólida para o desenvolvimento subsequente do sistema. Espera-se que a modelagem detalhada fornecida neste trabalho não apenas oriente a implementação do sistema, mas também promova uma compreensão compartilhada dos requisitos e funcionalidades entre os membros da equipe de desenvolvimento.

## 2. Diagrama de Casos

### Pessoa Física

- **Atores:** Aluno, Professor, Funcionário;
- **Objetivos:** Cadastrar, consultar, alterar e excluir dados de pessoas físicas no Sistema.

### Pessoa Jurídica

- **Atores:** Fornecedor
- **Objetivos:** Cadastrar, consultar, alterar e excluir dados de pessoas jurídicas no sistema.

### Professor

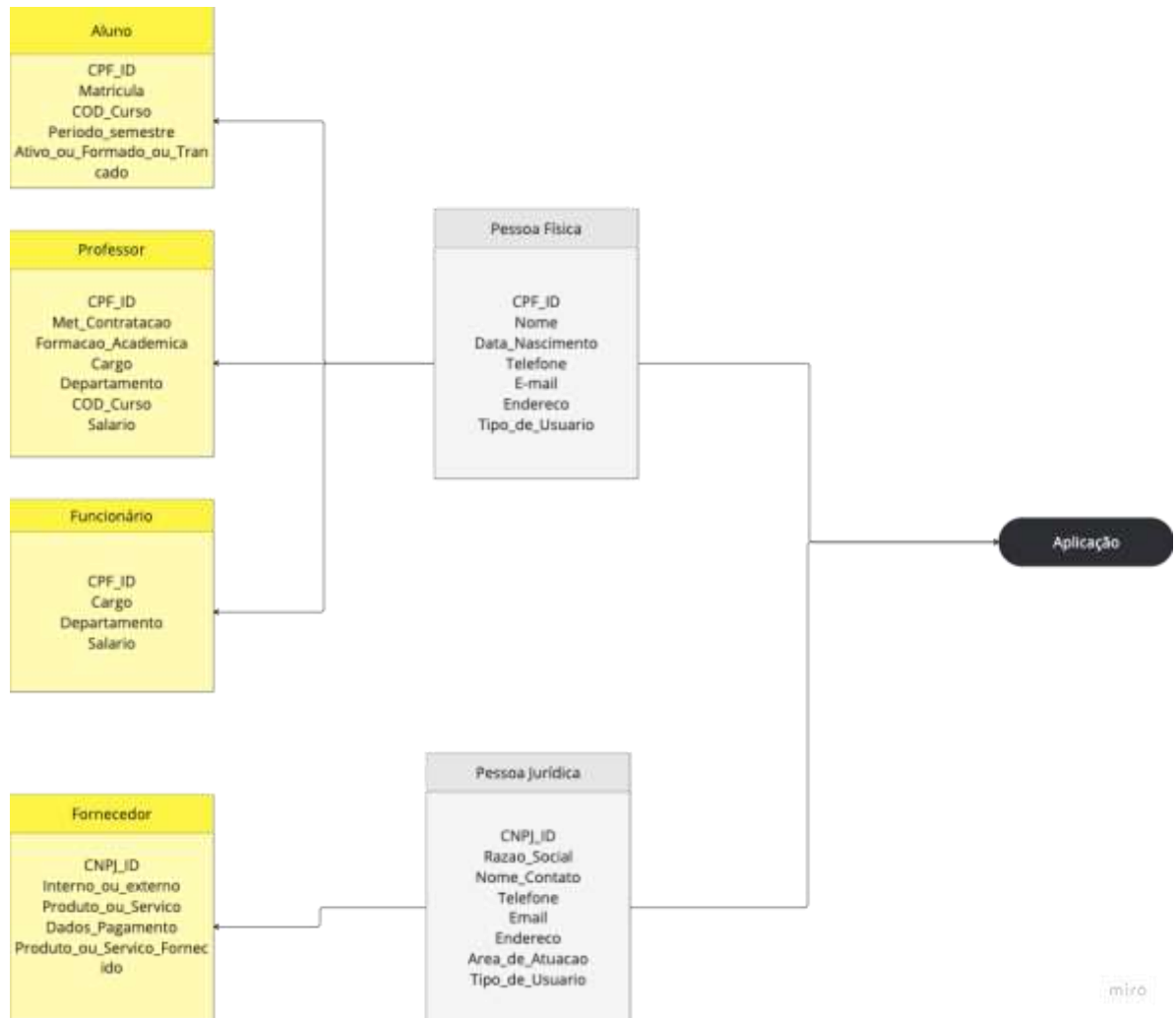
- Atores: Professor
- Objetivos: Cadastrar, consultar, alterar e excluir dados de professor no sistema.

### Fornecedor

- Atores: Fornecedor
- Objetivos: Cadastrar, consultar, alterar e excluir dados de fornecedor no sistema.

### Aluno

- Atores: Aluno
- Objetivos: Cadastrar, consultar, alterar e excluir dados de aluno no sistema.



### 3. Descrição de Cenários

#### **Matrícula de Aluno em Curso**

Cenário Principal:

Pré-condição: Aluno deve estar cadastrado e logado no sistema.

- Ações:
  1. Aluno faz login no sistema.
  2. Aluno busca e seleciona o curso de sua preferência.
  3. Aluno clica no botão 'Inscrever-se'.
  4. Sistema verifica a elegibilidade do aluno para o curso.

Pós-condição: Aluno é matriculado no curso e recebe confirmação da matrícula.

#### **Cenário Alternativo 1: Curso na Lotação Máxima**

- Pré-condição: Aluno deve estar logado e ter selecionado um curso.
- Condição: Curso escolhido está com lotação máxima.
- Resultado: Aluno é adicionado à lista de espera.
- Pós-condição: Aluno recebe notificação de que está na lista de espera do curso.

#### **Cenário Alternativo 2: Pré-Requisitos Não Atendidos**

- Pré-condição: Aluno deve estar logado e procurando cursos.
- Condição: Aluno não atende aos pré-requisitos do curso escolhido.
- Resultado: Sistema informa sobre os pré-requisitos.
- Pós-condição: Aluno pode buscar outros cursos ou obter as qualificações necessárias.

#### **Professor Criando um Curso**

Cenário Principal:

Pré-condição: Professor deve estar cadastrado e logado no sistema.

- Ações:
  1. Professor acessa a seção de gerenciamento de cursos.



2. Opta por criar um novo curso, inserindo as informações necessárias.
3. Sistema verifica as informações e confirma a criação do curso.

Pós-condição: Curso é criado e disponibilizado no sistema.

#### **Cenário Alternativo 1: Informações Incompletas**

- Pré-condição: Professor iniciou o processo de criação de curso.
- Condição: Campos essenciais estão incompletos.
- Resultado: Sistema solicita o preenchimento dos campos faltantes.
- Pós-condição: Curso não é criado até que todas as informações sejam fornecidas.

#### **Cenário Alternativo 2: Curso Existente**

- Pré-condição: Professor está criando um novo curso.
- Condição: Curso já existe no sistema.
- Resultado: Sistema notifica que o curso já existe.
- Pós-condição: Evita-se a duplicação de cursos no sistema.

### **Fornecedor Oferecendo Materiais**

#### **Cenário Principal:**

Pré-condição: Fornecedor deve estar cadastrado e logado no sistema.

- Ações:
  1. Fornecedor acessa a seção de pedidos.
  2. Escolhe um pedido e confirma o fornecimento.
  3. Sistema atualiza o status do pedido após confirmação de entrega.

Pós-condição: Materiais são fornecidos e o pedido é atualizado como concluído.

#### **Cenário Alternativo 1: Pedido Já Atendido**

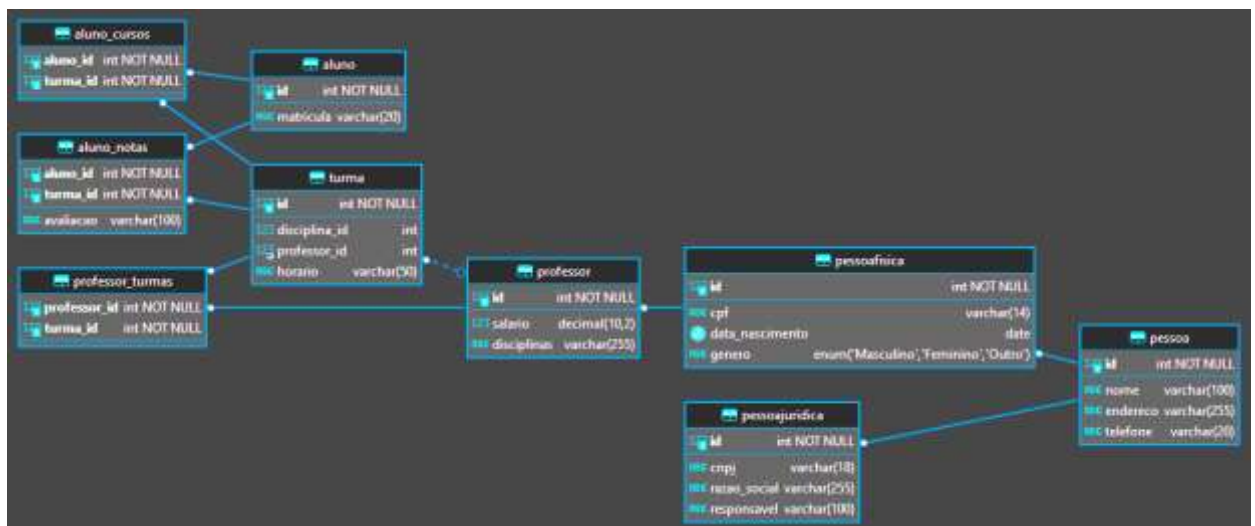
- Pré-condição: Fornecedor está tentando fornecer materiais para um pedido.
- Condição: Pedido já foi atendido.

- Resultado: Sistema notifica que o pedido já foi atendido.
- Pós-condição: Fornecedor é informado da situação e o pedido permanece concluído.

### Cenário Alternativo 2: Estoque Insuficiente

- Pré-condição: Fornecedor seleciona um pedido para fornecer materiais.
- Condição: Fornecedor tem estoque insuficiente para o pedido.
- Resultado: Sistema notifica sobre o estoque insuficiente.
- Pós-condição: Pedido não é atendido até que o fornecedor reabasteça o estoque.

## 4. Diagrama de Classe



## 5. Dump MySQL

```
-- MySQL dump 10.13 Distrib 8.0.19, for Win64 (x86_64)
--
-- Host: localhost Database: analyse_de_sistemas
--
-----
-- Server version 8.0.28

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `aluno`
--

DROP TABLE IF EXISTS `aluno`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `aluno` (
  `id` int NOT NULL AUTO_INCREMENT,
  `matricula` varchar(20) COLLATE utf8mb4_general_ci DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
```

```

-- Dumping data for table `aluno`
--

LOCK TABLES `aluno` WRITE;
/*!40000 ALTER TABLE `aluno` DISABLE KEYS */;
/*!40000 ALTER TABLE `aluno` ENABLE KEYS */;
UNLOCK TABLES;

--

-- Table structure for table `aluno_cursos`
--

DROP TABLE IF EXISTS `aluno_cursos`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `aluno_cursos` (
  `aluno_id` int NOT NULL,
  `turma_id` int NOT NULL,
  PRIMARY KEY (`aluno_id`,`turma_id`),
  KEY `turma_id` (`turma_id`),
  CONSTRAINT `aluno_cursos_ibfk_1` FOREIGN KEY (`aluno_id`) REFERENCES `aluno` (`id`),
  CONSTRAINT `aluno_cursos_ibfk_2` FOREIGN KEY (`turma_id`) REFERENCES `turma` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--

-- Dumping data for table `aluno_cursos`
--

LOCK TABLES `aluno_cursos` WRITE;
/*!40000 ALTER TABLE `aluno_cursos` DISABLE KEYS */;
/*!40000 ALTER TABLE `aluno_cursos` ENABLE KEYS */;
UNLOCK TABLES;

--

-- Table structure for table `aluno_notas`

```

--

DROP TABLE IF EXISTS `aluno\_notas`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!50503 SET character\_set\_client = utf8mb4 \*/;

CREATE TABLE `aluno\_notas` (

  `aluno\_id` int NOT NULL,

  `turma\_id` int NOT NULL,

  `avaliacao` varchar(100) COLLATE utf8mb4\_general\_ci DEFAULT NULL,

  PRIMARY KEY (`aluno\_id`, `turma\_id`),

  KEY `turma\_id` (`turma\_id`),

  CONSTRAINT `aluno\_notas\_ibfk\_1` FOREIGN KEY (`aluno\_id`) REFERENCES `aluno` (`id`),

  CONSTRAINT `aluno\_notas\_ibfk\_2` FOREIGN KEY (`turma\_id`) REFERENCES `turma` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `aluno\_notas`

--

LOCK TABLES `aluno\_notas` WRITE;

/\*!40000 ALTER TABLE `aluno\_notas` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `aluno\_notas` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `pessoa`

--

DROP TABLE IF EXISTS `pessoa`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!50503 SET character\_set\_client = utf8mb4 \*/;

CREATE TABLE `pessoa` (

  `id` int NOT NULL AUTO\_INCREMENT,

  `nome` varchar(100) COLLATE utf8mb4\_general\_ci DEFAULT NULL,

  `endereco` varchar(255) COLLATE utf8mb4\_general\_ci DEFAULT NULL,

```

`telefone` varchar(20) COLLATE utf8mb4_general_ci DEFAULT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `pessoa`
--

LOCK TABLES `pessoa` WRITE;
/*!40000 ALTER TABLE `pessoa` DISABLE KEYS */;
/*!40000 ALTER TABLE `pessoa` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `pessoafisica`
--

DROP TABLE IF EXISTS `pessoafisica`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `pessoafisica` (
  `id` int NOT NULL,
  `cpf` varchar(14) COLLATE utf8mb4_general_ci DEFAULT NULL,
  `data_nascimento` date DEFAULT NULL,
  `genero` enum('Masculino','Feminino','Outro') COLLATE utf8mb4_general_ci DEFAULT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `pessoafisica_ibfk_1` FOREIGN KEY (`id`) REFERENCES `pessoa` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `pessoafisica`
--

LOCK TABLES `pessoafisica` WRITE;

```

```

/*!40000 ALTER TABLE `pessoafisica` DISABLE KEYS */;
/*!40000 ALTER TABLE `pessoafisica` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `pessoajuridica`
--

DROP TABLE IF EXISTS `pessoajuridica`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `pessoajuridica` (
  `id` int NOT NULL,
  `cnpj` varchar(18) COLLATE utf8mb4_general_ci DEFAULT NULL,
  `razao_social` varchar(255) COLLATE utf8mb4_general_ci DEFAULT NULL,
  `responsavel` varchar(100) COLLATE utf8mb4_general_ci DEFAULT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `pessoajuridica_ibfk_1` FOREIGN KEY (`id`) REFERENCES `pessoa` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `pessoajuridica`
--

LOCK TABLES `pessoajuridica` WRITE;
/*!40000 ALTER TABLE `pessoajuridica` DISABLE KEYS */;
/*!40000 ALTER TABLE `pessoajuridica` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `professor`
--

DROP TABLE IF EXISTS `professor`;
/*!40101 SET @saved_cs_client = @@character_set_client */;

```

```

/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `professor` (
  `id` int NOT NULL,
  `salario` decimal(10,2) DEFAULT NULL,
  `disciplinas` varchar(255) COLLATE utf8mb4_general_ci DEFAULT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `professor_ibfk_1` FOREIGN KEY (`id`) REFERENCES `pessoafisica` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `professor`
--

LOCK TABLES `professor` WRITE;
/*!40000 ALTER TABLE `professor` DISABLE KEYS */;
/*!40000 ALTER TABLE `professor` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `professor_turmas`
--

DROP TABLE IF EXISTS `professor_turmas`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `professor_turmas` (
  `professor_id` int NOT NULL,
  `turma_id` int NOT NULL,
  PRIMARY KEY (`professor_id`,`turma_id`),
  KEY `turma_id` (`turma_id`),
  CONSTRAINT `professor_turmas_ibfk_1` FOREIGN KEY (`professor_id`) REFERENCES `professor` (`id`),
  CONSTRAINT `professor_turmas_ibfk_2` FOREIGN KEY (`turma_id`) REFERENCES `turma` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

```



```

--
-- Dumping data for table `professor_turmas`
--

LOCK TABLES `professor_turmas` WRITE;
/*!40000 ALTER TABLE `professor_turmas` DISABLE KEYS */;
/*!40000 ALTER TABLE `professor_turmas` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `turma`
--

DROP TABLE IF EXISTS `turma`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `turma` (
  `id` int NOT NULL AUTO_INCREMENT,
  `disciplina_id` int DEFAULT NULL,
  `professor_id` int DEFAULT NULL,
  `horario` varchar(50) COLLATE utf8mb4_general_ci DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `professor_id` (`professor_id`),
  CONSTRAINT `turma_ibfk_1` FOREIGN KEY (`professor_id`) REFERENCES `professor` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `turma`
--

LOCK TABLES `turma` WRITE;
/*!40000 ALTER TABLE `turma` DISABLE KEYS */;
/*!40000 ALTER TABLE `turma` ENABLE KEYS */;
UNLOCK TABLES;

```

```
--  
-- Dumping routines for database 'analise_de_sistemas'  
--  
/!*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;  
  
/!*!40101 SET SQL_MODE=@OLD_SQL_MODE */;  
/!*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;  
/!*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;  
/!*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
/!*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;  
/!*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;  
/!*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;  
  
-- Dump completed on 2024-03-23 16:51:02
```

## Conclusão

Ao longo deste Projeto Integrador, dedicamos esforços para a modelagem de um sistema de gestão escolar baseado na web, aplicando os conceitos e técnicas da Linguagem Unificada de Modelagem (UML). O objetivo principal foi representar de forma clara e precisa os requisitos, interações e estrutura de classes do sistema proposto, visando aprimorar a compreensão e comunicação entre os membros da equipe de desenvolvimento.

Por meio da elaboração de diagramas de casos de uso, descrições detalhadas de cenários de interação e um diagrama de classe abrangente, conseguimos capturar as principais funcionalidades e entidades do sistema, como cadastro de alunos, professores, fornecedores e gestão de cursos. Essa modelagem detalhada serviu como base sólida para o desenvolvimento subsequente do projeto, orientando a implementação das funcionalidades de forma eficiente e organizada.

Através dos cenários de interação, pudemos explorar diversas situações possíveis de uso do sistema, identificando casos principais e alternativos, pré-condições e pós-condições, o que contribuiu significativamente para a compreensão e validação dos requisitos do sistema.

Além disso, a criação do diagrama de classe nos permitiu visualizar a estrutura de dados necessária para suportar as operações do sistema, destacando as relações entre as entidades e suas respectivas propriedades e métodos.

Em suma, esta primeira etapa do Projeto Integrador foi fundamental para estabelecer uma base sólida e bem fundamentada para o desenvolvimento futuro do sistema de gestão escolar. Esperamos que a modelagem detalhada realizada neste trabalho contribua não apenas para a implementação bem-sucedida do sistema, mas também para uma compreensão compartilhada dos requisitos e funcionalidades entre os membros da equipe de desenvolvimento.

## Referências

MIRO. Disponível em: <https://miro.com/>. Acesso em: 18 mar. 2024.

OLIVEIRA, Henrique Pontes Gonçalves de. Análise de sistemas. São Paulo: Editora Senac São Paulo, 2019. (Série Universitária)