

MINISTÉRIO DA EDUCAÇÃO  
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
FARROUPILHA  
CAMPUS SANTO ÂNGELO

Sistema De Ajuste De Matrícula

André Gallas

Santo Ângelo, Rs, Brasil.  
2023

# Sistema De Ajuste De Matrícula

Por

André Gallas

Sistema visa a disponibilidade de uma plataforma on-line para o ajuste e gerenciamentos de matrícula no início dos semestres letivos do campus

Santo Ângelo, RS, Brasil.  
2023

MINISTÉRIO DA EDUCAÇÃO  
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
FARROUPILHA  
CAMPUS SANTO ÂNGELO

A Comissão Examinadora, abaixo assinada, Aprova a Monografia

Sistema De Ajuste De Matrícula

Elaborado por: André Gallas

**Como requisito parcial para a obtenção do título de Tecnólogo em Sistemas  
para Internet**

**COMISSÃO EXAMINADORA**

---

Juliano Gomes Weber  
**(Professor/Orientador)**

---

Marcos Vinicius Pizzutti Nascimento  
**Instituto Federal Farroupilha campus Santo Ângelo**

---

Allan Motta Schumacher  
**Instituto Federal Farroupilha campus Santo Ângelo**

Conceito Final:.....

Santo Ângelo.....de.....de 2023

## **RESUMO**

Com a popularização da internet e de dispositivos móveis com bom desempenho e aumento de suas funcionalidades, houve um aumento de demanda por sistemas disponíveis via web. A disponibilidade de um sistema online aumenta o conforto e amplia o acesso, além de concentrar as informações necessárias ao andamento dos processos. Existe uma carência de uma plataforma que possibilite o ajuste de matrículas por parte dos alunos bem como a possibilidade de acompanhar o andamento e o progresso das solicitações pela coordenação de curso e pela secretarias de registros acadêmicos. Infelizmente a integração com o Sistema já vigente na instituição não foi possível devido a falta de disponibilidade dos desenvolvedores do mesmo, porém foi pensada uma forma de importação via arquivos externos. O sistema conta com formas de cadastro de alunos, coordenadores, servidores, bem como formulários e agendamentos.

**Palavras-chave:** Ajuste de matrícula, importação, agendamentos, formulários.

## Abstract

With the popularization of the internet and mobile devices with good performance and increased functionality, there has been an increase in demand for systems available via the web. The availability of an online system increases comfort and expands access, in addition to concentrating the information necessary for the progress of processes. There is a lack of a platform that allows students to adjust enrollments as well as the possibility of monitoring the progress and progress of requests by the course coordinator and the academic records office. Unfortunately, integration with the System already in force at the institution was not possible due to the lack of availability of its developers, but a way of importing via external files was devised. The system has forms for registering students, coordinators, servers, as well as forms and schedules.

## **LISTAS DE ABREVIATURAS**

SIGAA - Sistema Integrado de Gestão de Atividades Acadêmica

REST - Representational State Transfer

CSV - Comma-Separated Values

HTML - Hyper Text Markup Language

HTTP - Hypertext Transfer Protocol

HTTPS - Hypertext Transfer Protocol Secure

SQL - Structured Query Language

SGBDR - Sistema Gerenciador De Banco De Dados Relacional

JSON - JavaScript Object Notation

XML - Extensible Markup Language

MVC - Model View Controller

ORM - Object Relational Mapping

JSF - Java Server Faces

AJAX - Asynchronous JavaScript and XML

ID - identifier

DRA - Diretoria de Registros Acadêmicos

# SUMÁRIO

<b>RESUMO</b> .....	<b>4</b>
<b>LISTAS DE ABREVIATURAS</b> .....	<b>6</b>
<b>SUMÁRIO</b> .....	<b>7</b>
<b>1 INTRODUÇÃO</b> .....	<b>8</b>
1.1 Objetivos.....	8
1.1.1 Objetivo Geral.....	9
1.1.2 Objetivos Específicos.....	9
1.2 Trabalhos Correlatos.....	9
1.2.1 Integração SIGAA com plataforma SOLAR.....	9
1.2.2 Integração entre SIGAA e a Ferramenta Multiprova.....	10
1.2.3 Importação de dados a partir de Arquivos CSV.....	11
1.3 FUNDAMENTAÇÃO TEÓRICA.....	14
1.3.1 Java.....	14
1.3.2 Protocolos HTTP/HTTPS.....	14
1.3.3 SQL.....	15
1.3.4 JSON.....	15
1.3.5 XML.....	15
1.3.6 REST.....	16
1.3.7 CSV.....	16
1.3.8 MVC.....	16
1.3.9 ORM.....	16
1.3.10 HIBERNATE.....	17
1.3.11 JSF.....	17
1.3.12 AJAX.....	17
1.3.13 PRIMEFACES.....	18
<b>2 DESENVOLVIMENTO</b> .....	<b>18</b>
2.1 CASOS DE USO.....	19
2.1.1 DIAGRAMA DE CASOS DE USO.....	19
2.1.2 ESPECIFICAÇÃO DE CASOS DE USO.....	20
2.2 BANCO DE DADOS.....	26
2.2.1 DIAGRAMA ENTIDADE RELACIONAMENTO.....	26
2.2.2 MODELO LÓGICO.....	27
2.3 DIAGRAMA DE CLASSES.....	28
2.4 TELAS DO SISTEMA.....	29
<b>CONCLUSÃO</b> .....	<b>33</b>
<b>REFERÊNCIAS</b> .....	<b>34</b>

# 1 INTRODUÇÃO

Sistemas web facilitam o acesso e troca de informações entre usuário final e as organizações, centralizam e facilitam o processamento dos dados além da disponibilidade em qualquer lugar com acesso a internet.

A integração entre sistemas também é uma realidade, sendo essa uma das principais demandas da área de tecnologia atualmente, novos produtos, serviços e demandas para interação com sistemas já existentes surgem à medida que o mercado se diversifica e se amplia.

Demanda por arquivamento de documentos também tem sofrido pressão para que sejam digitalizados, já que formulários de papel podem conter erros, rasuras ou informações conflitantes e até mesmo incompletas, com isso o arquivamento e controle de processos internos de forma digital é benéfico para as partes envolvidas, mais uma vez com vistas na praticidade e facilidade de acesso e manutenção dos documentos, além do benefício ao meio ambiente.

O IFFar, como toda instituição pública e, no caso desta, de ensino tem processos internos que auxiliam a organização das atividades e demandas dos alunos e das coordenações de curso bem como das secretarias, alguns processos ainda são baseados em formulários padrão de papel, como ajustes e cancelamentos de matrículas e validação de componentes curriculares.

Sendo assim, o trabalho se propõe a elaborar e implementar um sistema web que possibilite o acesso a formulários de quebra de pré-requisitos e de validação e ajuste de componentes curriculares, utilizados pela secretaria de Registros acadêmicos do campus e que ainda não são digitais. A plataforma pretende integrar os dados com o SIGAA, primeiramente a idéia era de que fosse possível a integração direta com uso de API ou conexão via REST, mais tarde resolveu-se utilizar importação por arquivos .csv para entrar com os dados de usuário e disciplina. Também é proposta do sistema permitir ao aluno acompanhar o andamento e status de seu pedido.

## 1.1 Objetivos

### 1.1.1 Objetivo Geral

Desenvolver uma solução de software que possibilite a digitalização de processos organizacionais internos do IFFar Campus Santo Ângelo, referentes a ajustes de matrícula e disciplinas dos alunos, que são atualmente utilizados exclusivamente em papel.

Os benefícios da digitalização dos processos são conhecidos como a praticidade em gerenciar os processos, facilidade no manuseio dos dados bem como no acompanhamento do andamento dos pedidos.

### 1.1.2 Objetivos Específicos

- Possibilitar o acesso e realização de ajustes de matrícula via web, de forma facilitada;
- Facilitar o gerenciamento dos processos de ajuste pelos servidores do campus;
- Auxiliar a coordenação de curso no acompanhamento dos processos, quanto a períodos, disponibilidade e status dos processos de ajuste de matrícula;
- Vincular autenticação integrada com o Google para facilitar o acesso à plataforma;
- Agilizar os processos de ajuste de matrícula dentro da instituição, possibilitando a digitalização dos processos, e melhorando o controle dos processos tanto por parte dos alunos quanto dos servidores;

## 1.2 Trabalhos Correlatos

### 1.2.1 Integração SIGAA com plataforma SOLAR

Quando se deu a pandemia, surgiu a necessidade de utilização de ambientes virtuais e aprendizagem, para suprir essa demanda a Universidade Federal do Ceará UFC, utilizou a plataforma SOLAR, que permite a gestão de conteúdo às turmas virtuais bem como agendamento de teleconferência, frequência dos alunos e demais necessidades do professor.

Surgiu então a necessidade de integração entre essa nova plataforma de aprendizagem com o SIGAA, e o projeto da instituição permitiu que os professores conseguissem importar as turmas diretamente do SIGAA para o SOLAR, trazendo os dados da turma, como código, descrição e participantes.

Figura 1: Tela do Sistema SOLAR

The screenshot displays the SOLAR system interface. At the top, there is a navigation bar with the SOLAR logo, a user profile for 'Professor', the time '13:05:05', and a 'Assistir' button. Below this, a breadcrumb trail shows 'HOME > QUÍMICA I - 2011.1'. The main content area is titled 'Licenciatura em Química - Química I - 2011.1' and includes a 'Turma' label. A table lists the course details:

Disciplina	Responsáveis
Pensando mais a longo prazo, a percepção das dificuldades não causa impacto indireto na reavaliação da fórmula da ressonância racionalista.	Aluno 3 (Aluno Monitor) ✉ Professor (Prof. Titular, Prof. Titular UAB, Tutor a... ✉ Tutor a Dist (Tutor a Distância) ✉ Tutor Presencial (Tutor Presencial) ✉

Below the table, there is a section for 'Aulas' (Classes) with a list of modules and dates:

- modulo 1
  - 05/03/2011 - 20/06/2011 [aula 1 pag ufc](#)
  - 27/03/2011 - 27/06/2011 [aula 2 pag uol](#)
- modulo 2
  - 25/03/2011 - 27/08/2011 [aula 3](#)
  - 16/03/2011 - 01/08/2012 [aula 4](#)

At the bottom, there are sections for 'Mensagens' (Messages) showing 'Sem novas mensagens', 'Fórum' (Forum) with '25/02 User 2 As experiências', and 'Agenda' (Calendar) for 'Março 2020'.

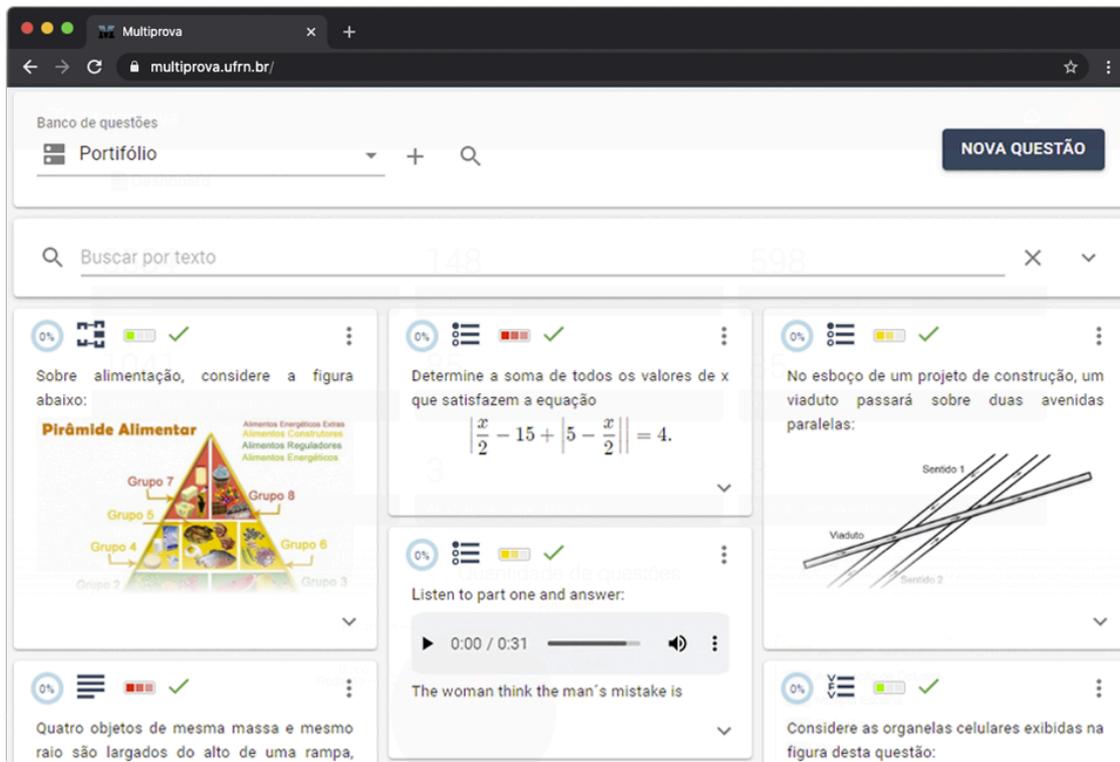
Fonte: <https://www.ufc.br/noticias/14434-instituto-ufc-virtual-e-sti-disponibilizam-tutoriais-de-integracao-do-sigaa-com-o-solar-e-de-usos-dos-sistemas>, acesso em 2 Março de 2023

### 1.2.2 Integração entre SIGAA e a Ferramenta Multiprova

A ferramenta Multiprova permite avaliar as condições de ensino-aprendizagem de alunos por meio de avaliações periódicas. Essa ferramenta permite o gerenciamento de questões aplicadas ao aluno, uma vez que é possível predefinir uma série de alterações em uma única questão sem alterar a dificuldade de resolução da mesma, ou seja, alterando aspectos acessórios da questão.

A integração entre os dois sistemas foi baseada em RESTful e propôs a utilização de informações do banco de dados do SIGAA como turmas, disciplinas, alunos e notas por parte do Multiprova, eliminando a necessidade de manutenção manual dessas informações entre ambos, além de garantir a unicidade dos dados.

Figura 2: Tela Sistema Multiprova



Fonte: <https://site.multiprova.ufrn.br/>, acesso em 2 de Março de 2023

### 1.2.3 Importação de dados a partir de Arquivos CSV

Se a integração entre o SIGAA e esse sistema proposto falhar, então será proposto a importação de dados a partir de um arquivo externo e posterior gravação em um banco de dados próprio, não é o principal objetivo dessa proposta, mas uma espécie de plano B.

Para permitir a importação, será necessário a utilização de um formato de arquivo simples e de código aberto, como o .CSV, que organiza os dados em linhas, delimitando cada informação por um separador padrão, geralmente a informação está contida entre aspas duplas.

É possível a leitura desse tipo de arquivo de acordo com a documentação oficial do projeto [opencsv](https://opencsv.org/) e demais fontes não oficiais e não documentadas.

Figura 3: Código para processamento de arquivos CSV

```

public class TextToTeacher extends AbstractCsvConverter {

    @Override
    public Object convertToRead(String value) {
        Teacher t = new Teacher();
        String[] split = value.split("\\.", 2);
        t.setSalutation(split[0]);
        t.setSurname(split[1]);
        return t;
    }

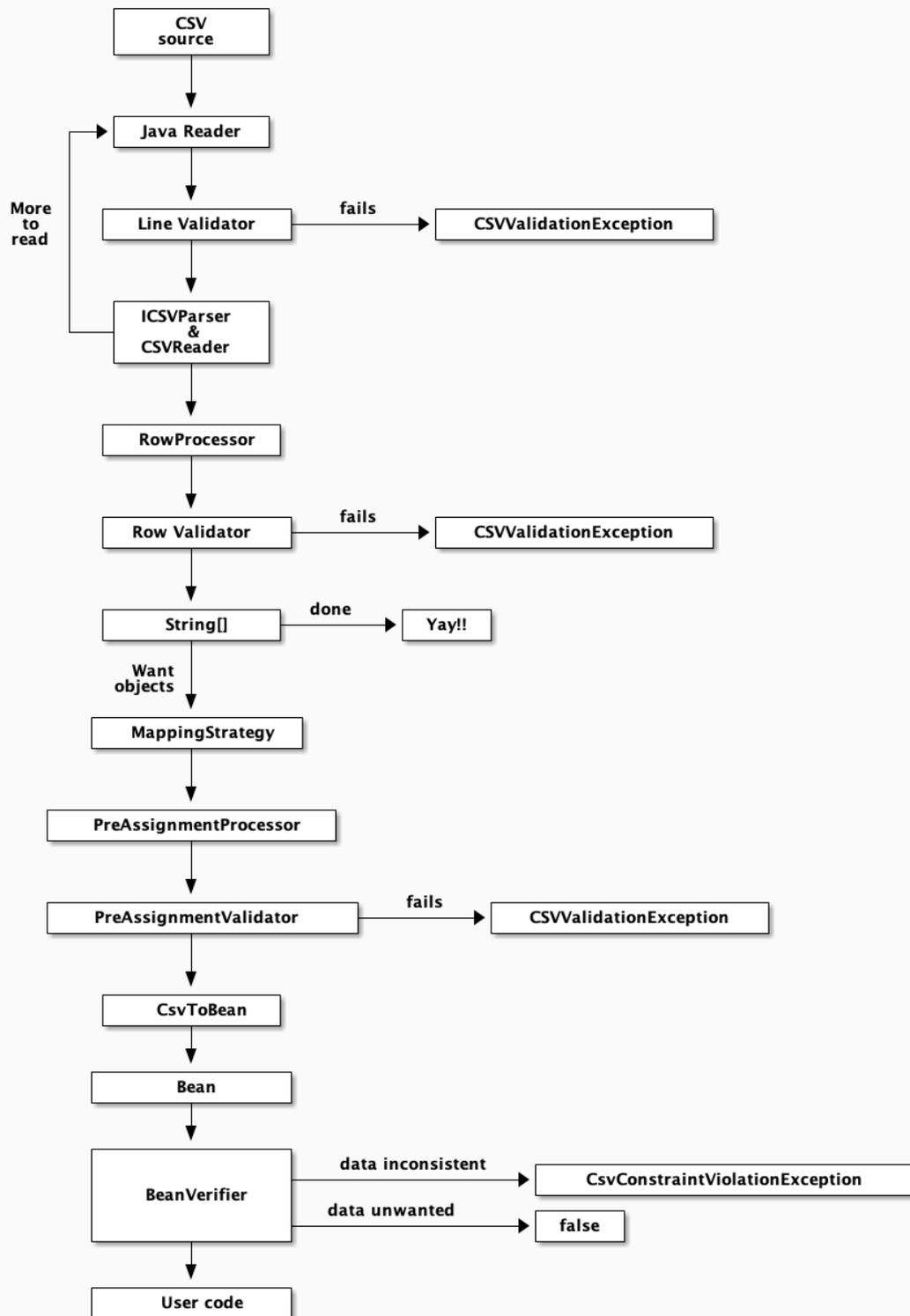
    @Override
    public String convertToWrite(Object value) {
        Teacher t = (Teacher) value;
        return String.format("%s.%s", t.getSalutation(), t.getSurname());
    }
}

```

Fonte: <https://opencsv.sourceforge.net/> acesso em 13 de novembro de 2023.

Esse é apenas um exemplo que mostra a manipulação do conteúdo linha a linha tomando o cuidado de criar células através do identificador padrão, que pode ser vírgula, dois pontos ou ponto e vírgula.

Figura 3: Esquema de interpretação básico de arquivo CSV



Fonte: <https://opencsv.sourceforge.net/#reading>, acesso em 2 de março de 2023.

## 1.3 FUNDAMENTAÇÃO TEÓRICA

### 1.3.1 Java

Em 1991, a Sun Microsystems, percebendo o potencial que os microchips trariam para dispositivos de computação, criou um projeto interno chamado Green Project Tendo James Gosling um dos criadores da linguagem inicialmente apelidada de OAK, porém por motivos de patente não foi possível utilizar essa nomenclatura, sendo alterada para Java. O projeto buscava mostrar a percepção de integração interplataformas que a empresa tinha ao perceber que os dispositivos se intercomunicariam em questão de tempo, então foi demonstrado um aparelho chamado de \*7(star seven) com um assistente virtual chamado Duke, que tinha como função orientar o usuário por um tour pelo aparelho. Java é uma linguagem de programação orientada a objetos inicialmente baseada em C e C++.

A chave do Java, é sua versatilidade, pois permite utilizar o mesmo programa em uma variedade muito grande de dispositivos, já que executa os código sobre uma espécie de máquina virtual que faz o trabalho de compatibilidade entre a plataforma e a aplicação. Isso é geralmente chamado de “escreva uma vez, execute em qualquer lugar”

Em 1993, com a explosão da *Web*, a Sun viu a possibilidade de levar conteúdo dinâmico aos usuários através do Java. A comunidade empresarial também viu o potencial da linguagem para utilização na *Web*, ela é agora utilizada para desenvolver aplicativos para computadores e uma variedade de outros dispositivos, além de ser executada em servidores que fornecem serviços ao usuário final.

### 1.3.2 Protocolos HTTP/HTTPS

Hyper Text Transfer Protocol, é o protocolo utilizado para acesso de páginas de internet por clientes na *Web*, é baseado em requisições por parte dos clientes e respostas por parte dos servidores.

### 1.3.3 SQL

Structured Query Language (Linguagem de Consulta Estruturada), é a linguagem padrão para Sistemas Gerenciadores de Banco de Dados Relacionais. Esse padrão foi estabelecido para facilitar a implementação e migração de bancos de dados entre Sistemas gerenciadores de Bancos de Dados Relacionais SGBDRs.

Segundo ELMASRI, R,

SQL é uma linguagem de banco de dados abrangente: tem instruções para definição de dados, consultas e atualizações. Logo, ela é uma DDL e uma DML. Além disso, ela tem facilidades para definir visões sobre o banco de dados, para especificar segurança e autorização, para definir restrições de integridade e para especificar controles de transação. Ela também possui regras para embutir instruções SQL em uma linguagem de programação de uso geral.

Os padrões SQL mais recentes, são divididos em uma especificação núcleo mais extensões especializadas. O núcleo compõe um conjunto comum de comandos e capacidades aos SGBDRs , já as extensões especializadas são um conjunto de bibliotecas, e adendos otimizados para cada SGBDR

### 1.3.4 JSON

JavaScript Object Notation é um formato de troca de dados entre aplicações, tem como vantagem ser leve, facilmente legível e interpretável tanto por humanos quanto por máquinas, e possibilitar a troca de informações entre aplicações escritas em diferentes linguagens de programação.

JSON é implementado de duas formas, como um par nome/valor, e assim pode ser interpretado como objeto, lista encadeada, array, matriz associativa entre outros. Ou então como uma sequência de ordenada de valores, assim pode ser interpretado como array, vetor, lista ou sequência.

### 1.3.5 XML

Extensible Markup Language é uma linguagem de marcação de texto, utilizada para representar documentos estruturados, muito usado em sistemas Web. Permite que através de tags seja possível organizar a informação com padrões bem estabelecidos de marcação.

### 1.3.6 REST

Representational State Transfer, é um estilo de arquitetura de software usado em web services que permite a troca de informações de maneira segura via protocolo HTTP e HTTPS, e retorna dados geralmente no formato XML ou JSON.

O termo RESTful é usado para indicar que um sistema suporta o conjunto de recursos REST, como os métodos GET para consulta de dados, POST para gravação, DELETE para exclusão e PUT para atualizar dados.

Outras funcionalidades essenciais do REST são a necessidade de cada requisição retornar de maneira independente e não deve ter estado, ou seja, precisa estar sempre disponível para acesso, para isso é usado o protocolo HTTP. Ter um conjunto de operações padronizadas.

Também se faz necessário a disponibilização de uma URI (Uniform Resource Identifier), esse será o endereço utilizado para troca de informações sobre o recurso.

### 1.3.7 CSV

Comma-Separated Values, é um formato de arquivo de texto puro em que os dados são organizados em forma de tabela. Valores de células podem ser colocados entre aspas duplas e cada célula é separada por vírgula, cada nova entrada é colocada logo abaixo da anterior, então cada linha do arquivo de texto corresponde a uma linha da tabela. Cada linha deve conter o mesmo número de células

### 1.3.8 MVC

Model View Controller é um padrão de projeto em que a aplicação é dividida em três camadas, a *View* que consiste na camada que usuário fará a interação, *controller* que recebe as requisições da view e as encaminha para a *model* se necessário, a camada *model* é responsável por conter classes das regras de negócio além de sus métodos.

### 1.3.9 ORM

Object/Relational Mapper é um framework que gerencia os dados de classes afim de gravá-los no banco de dados, a intenção é garantir a persistência das

informações. É uma ferramenta que auxiliar programadores na tarefa de gravação de informações em bancos de dados relacionais,

#### 1.3.10 HIBERNATE

Hibernate é um framework que se encarrega da recuperação de dados em aplicações java e faz isso utilizando as classes do java e tipos de dados para gerenciar automaticamente o banco de dados relacional. Ou seja, é possível a criação e atualização das tabelas através da edição dos atributos dos objetos em java, poupando uma boa parte do trabalho de desenvolvimento.

#### 1.3.11 JSF

Java Server Faces é um framework que facilita o gerenciamento de interfaces de usuário UIs em sistemas Java Web. Facilita o desenvolvimento das seguintes formas:

- Facilita a construção de uma UI usando um conjunto de componentes de IU reutilizáveis
- Simplifica a migração de dados da aplicação para a IU e provenientes dela
- Ajuda a gerenciar o estado da IU nas solicitações do servidor
- Oferece um modelo simples para conectar os eventos gerados pelo cliente ao código da aplicação do servidor
- Permite personalizar os componentes de UI para que sejam facilmente construídos e reutilizados

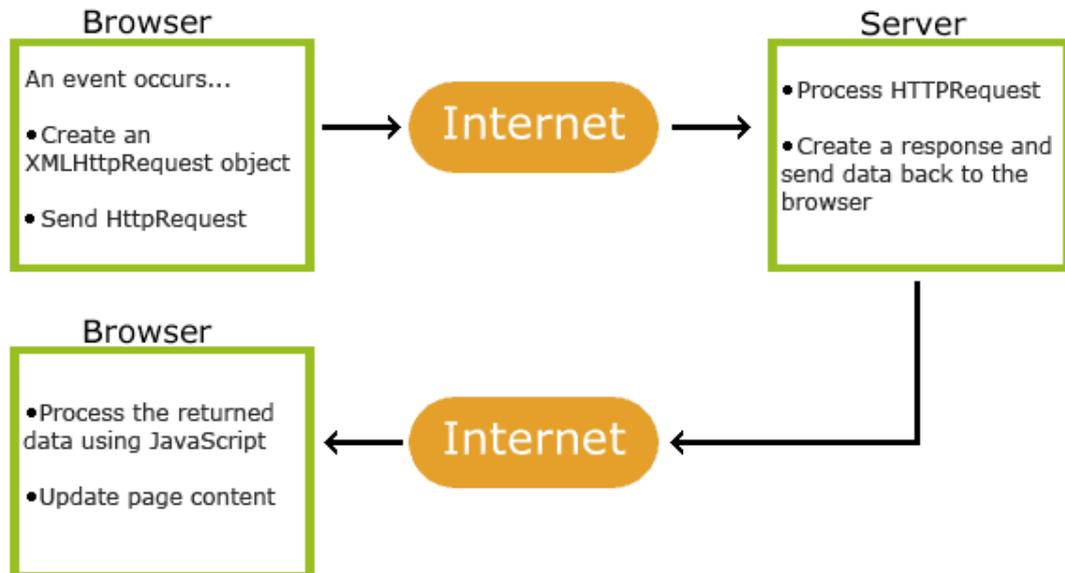
O código JSF é compilado antes de chegar ao usuário, e mantém os vínculos dos valores e objetos na aplicação, assim o usuário não tem acesso a nenhuma parte da lógica ou de funcionalidades internas da aplicação.

#### 1.3.12 AJAX

Asynchronous JavaScript And XML, não é uma linguagem de programação, mas faz uso de uma combinação de funcionalidades, como uma engine XMLHttpRequest object, que solicita dados ao servidor e JavaScript HTML DOM (Document Object Model) que permite alterar completamente qualquer elemento da

página html. Dessa forma o comportamento e conteúdo da página podem ser alterados conforme as solicitações e respostas da aplicação.

Figura 4: Funcionamento do AJAX



Fonte: [https://www.w3schools.com/whatis/whatis\\_ajax.asp](https://www.w3schools.com/whatis/whatis_ajax.asp) acesso em 16 de Março de 2023

### 1.3.13 PRIMEFACES

É um componente do JSF que gera a interface de usuário para uma aplicação web através de uma linguagem específica. Consiste em uma biblioteca que permite a geração de um layout específico incorporando segundo os autores do projeto: uma grande/rica variedade de componentes (HtmlEditor, Dialog, AutoComplete, Charts e muito mais), AJAX integrado baseado em APIs JSF AJAX padrão, aplicação leve, sem dependências, apenas um pacote jar utilizado na compilação do projeto, além de uma gama de facilidades incorporadas.

## 2 DESENVOLVIMENTO

O levantamento de requisitos do sistema resulta em uma série de diagramas que auxiliam o desenvolvimento da aplicação, mostra a estrutura organizacional básica do sistema organizando e evidenciando as demandas dos usuários, o modelo de negócio e a estrutura que será usada pelos desenvolvedores, descreve de forma clara as necessidades e os problemas que a aplicação atenderá.

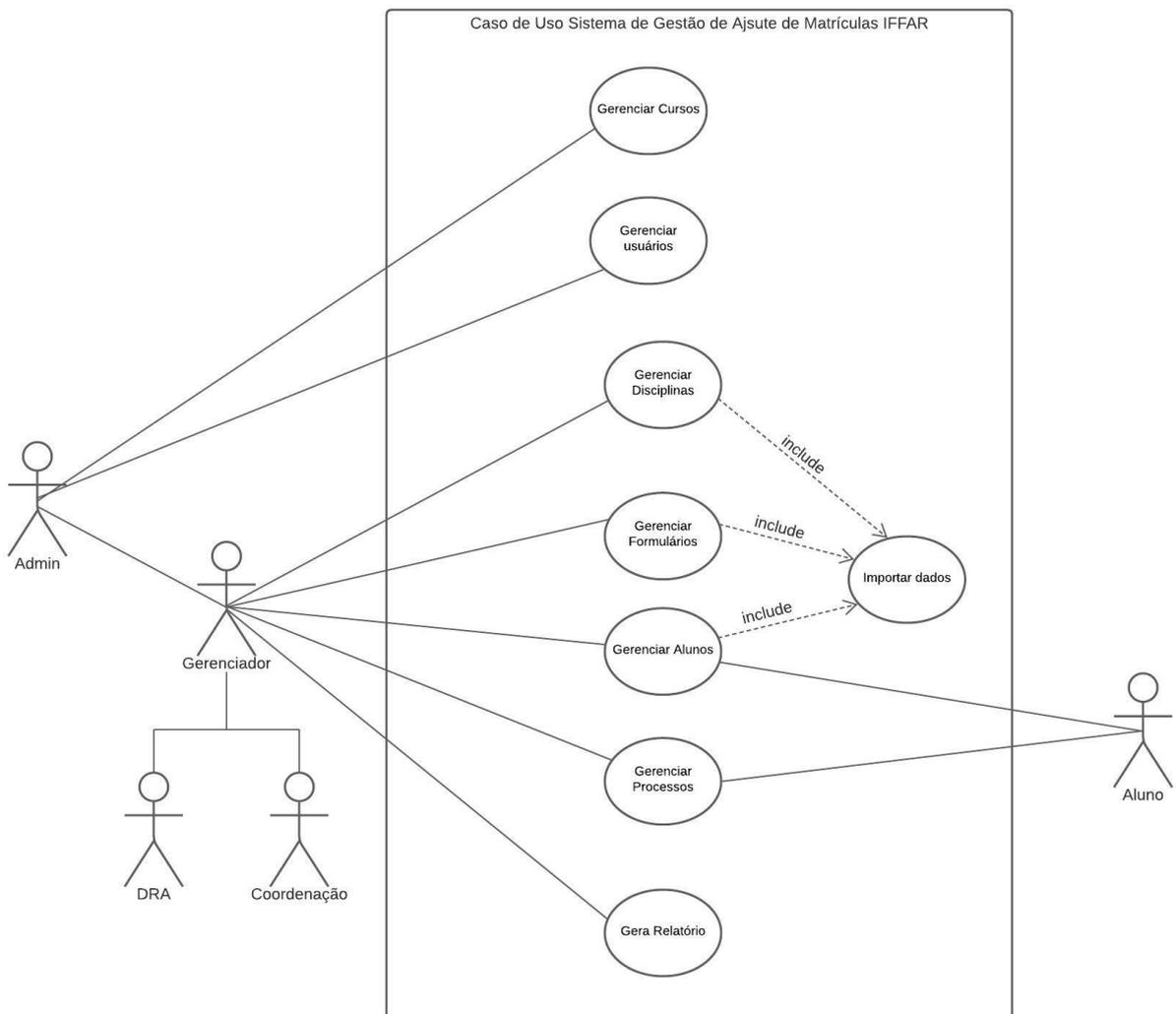
## 2.1 CASOS DE USO

Caso de uso é uma forma de representar o sistema de maneira abstrata com linguagem de alto nível, buscando esclarecer o papel dos atores na aplicação, demandas e os requisitos dos usuários que levaram à modelagem da aplicação com ela se apresentará depois de desenvolvida e utilizada.

### 2.1.1 DIAGRAMA DE CASOS DE USO

O Diagrama de casos de uso demonstra os atores do sistema e suas funcionalidades, é possível observar o acesso que cada usuário ou sistema terá sobre a aplicação.

Figura 5: Diagrama de casos de Uso



Fonte: Própria

## 2.1.2 ESPECIFICAÇÃO DE CASOS DE USO

IDENTIFICAÇÃO DO CASO DE USO	NOME DO CASO DE USO
UC1	Gerenciar Cursos
ATOR PRINCIPAL	ATOR SECUNDÁRIO
Admin	
PRÉ CONDIÇÕES	PÓS CONDIÇÕES
Login ativo no sistema	
FLUXO PRINCIPAL	
AÇÕES DO ATOR	AÇÕES DO SISTEMA
Adicionar e editar Informações referente ao curso, vinculado à coordenação	
FLUXO ALTERNATIVO I	
FLUXO ALTERNATIVO II	

Permite a adição de novos cursos, vinculação com os usuários com acesso, como a coordenação do curso respectivo. Não permite a exclusão de cursos, apenas a edição dos dados e vínculos. As atividades precisam ser mantidas em histórico (log) para futuras consultas e verificação de atividades.

IDENTIFICAÇÃO DO CASO DE USO	NOME DO CASO DE USO
UC2	Gerenciar Usuários
ATOR PRINCIPAL	ATOR SECUNDÁRIO
Admin	
PRÉ CONDIÇÕES	PÓS CONDIÇÕES
Autenticação via login	
FLUXO PRINCIPAL	
AÇÕES DO ATOR	AÇÕES DO SISTEMA
Adicionar e editar dados dos usuários	
FLUXO ALTERNATIVO I	
FLUXO ALTERNATIVO II	

Gerenciamento de usuários fornece uma interface que permite ao administrador do sistema a adição de novos usuários com privilégios de gerenciador, como coordenação dos cursos e usuários da DRA e demais setores, além da vinculação dos mesmos a seus setores e cursos.

A edição dos dados dos usuários inclui contato, nome, e curso ao qual o usuário está vinculado, bem como o setor. Ao alterar o curso ou setor de alocação do usuário, suas atividades anteriores com a função antiga precisam ser mantidas para consulta posterior, formando assim um histórico das atividades de cada usuário no sistema.

IDENTIFICAÇÃO DO CASO DE USO	NOME DO CASO DE USO
UC3	Gerenciar Disciplinas
ATOR PRINCIPAL	ATOR SECUNDÁRIO
Gerenciador	
PRÉ CONDIÇÕES	PÓS CONDIÇÕES
Autenticação via login	
FLUXO PRINCIPAL	
AÇÕES DO ATOR	AÇÕES DO SISTEMA
Adicionar, editar e excluir disciplinas do curso correspondente	
FLUXO ALTERNATIVO I	
FLUXO ALTERNATIVO II	

O gerenciamento de disciplinas fornece uma interface que permite a adição de disciplinas ministradas nos cursos de graduação, o cadastro de uma disciplina inclui sua ID do sistema SIGAA, descrição conforme mesmo sistema, curso(s) que a ofertam, semestre e carga horária. A edição da disciplina permite alterar qualquer valor utilizado no cadastro, porém a ID do SIGAA deve ser única, como é no sistema.

Exclusão de disciplina é possível, porém deve permanecer disponível para consulta e não pode haver perda de dados quando utilizada em algum formulário de aluno por exemplo.

IDENTIFICAÇÃO DO CASO DE USO	NOME DO CASO DE USO
UC4	Gerenciar Formulários
ATOR PRINCIPAL	ATOR SECUNDÁRIO
Gerenciador	
PRÉ CONDIÇÕES	PÓS CONDIÇÕES
Autenticação via login	
FLUXO PRINCIPAL	
AÇÕES DO ATOR	AÇÕES DO SISTEMA
Adicionar e editar formulários e seus tipos de acordo com o curso	
FLUXO ALTERNATIVO I	
FLUXO ALTERNATIVO II	

O gerenciamento de formulários fornece uma interface que permite o cadastro de novos formulários de preenchimento para solicitação e utilização por parte dos alunos. Os formulários podem ser adicionados pelos gerenciadores do sistema ou administrador, conforme diagrama de casos de uso acima. Formulário deve estar em conformidade com os formulários já existentes e utilizados na organização interna do campus, conforme resoluções internas. Deve coletar os dados dos alunos de maneira que permita sua identificação, cada formulário tem uma finalidade específica, então é necessário que exista um formulário por tipo de solicitação, como cancelamento de matrícula, trancamento, etc.. A edição de formulários permite a alteração de qualquer campo ou descrição utilizada e/ou disponível para os usuários do sistema. Os modelos de formulários estão disponíveis no site da instituição.

O aluno ao acessar o formulário poderá selecionar em uma lista as opções que deseja, como disciplinas que deseja validar ou cancelar, formulários de cancelamento ou validação que necessitem de upload de arquivo externo como comprovante de conclusão em outra instituição, deve estar disponível. O Aluno não pode validar disciplinas já cursadas ou que não estejam sendo ofertadas.

IDENTIFICAÇÃO DO CASO DE USO	NOME DO CASO DE USO
UC5	Gerenciar Alunos
ATOR PRINCIPAL	ATOR SECUNDÁRIO
Gerenciador	Aluno
PRÉ CONDIÇÕES	PÓS CONDIÇÕES
Autenticação via login	
FLUXO PRINCIPAL	
AÇÕES DO ATOR	AÇÕES DO SISTEMA
Adicionar e editar alunos vinculados aos cursos	
FLUXO ALTERNATIVO I	
FLUXO ALTERNATIVO II	

O usuário gerenciador tem acesso ao cadastro de alunos, informando os dados necessários à sua identificação como nome, matrícula, curso e afins, a matrícula precisa corresponder com o sistema SIGAA. Por definição, caso um aluno troque de curso, é gerada uma nova matrícula no SIGAA, então não pode haver o mesmo aluno com a mesma matrícula habilitado em dois cursos, porém o mesmo aluno pode cursar dois cursos, mas então com matrículas diferentes.

O aluno tem acesso a essa parte do sistema, mas poderá alterar a foto de perfil e dados de contato apenas, além de sua senha de acesso caso não seja possível a integração com o sistema SIGAA.

IDENTIFICAÇÃO DO CASO DE USO	NOME DO CASO DE USO
UC6	Gerenciar Processos
ATOR PRINCIPAL	ATOR SECUNDÁRIO
Gerenciador	Aluno
PRÉ CONDIÇÕES	PÓS CONDIÇÕES
Autenticação via login	
FLUXO PRINCIPAL	
AÇÕES DO ATOR	AÇÕES DO SISTEMA

Adicionar e editar processos baseados nos formulários e nos alunos, permite o acompanhamento do processo	
FLUXO ALTERNATIVO I	
FLUXO ALTERNATIVO II	

Após a inserção de alunos, disciplinas e formulários, será possível a criação de novos processos através da habilitação do acesso aos formulários por parte dos alunos de acordo com prazos estabelecidos por usuários gerenciadores.

Cada aluno acessa o sistema e seleciona qual ou quais processo(s) deseja iniciar selecionando um formulário em uma lista, os dados do aluno já estão preenchidos, visto que o aluno já está cadastrado, e seleciona disciplinas ou demais informações necessárias ao andamento de cada processo, como por exemplo ao validar uma matéria cursada em outra instituição, será necessário anexar documentos que comprovem a conclusão da disciplina..

Usuários gerenciadores têm autoridade para alterar o status dos processos, coordenação de curso pode cadastrar, habilitar e desabilitar o acesso aos processos por parte dos alunos de forma automática por meio do agendamento de disponibilidade de pedidos, ou seja, é possível estipular um prazo para que os formulários estejam disponíveis para preenchimento dos alunos.. Também pode alterar o status de andamento do processo como assinalando se a documentação está completa e se o processo foi concluído e/ou encaminhado ao setor responsável.

Usuários gerenciadores de outros setores como DRA podem alterar o status dos processos adicionando as informações do andamento do mesmo, como documentação insuficiente, ou mesmo deferido e assim dando o processo como concluído.

IDENTIFICAÇÃO DO CASO DE USO	NOME DO CASO DE USO
UC7	Importar dados
ATOR PRINCIPAL	ATOR SECUNDÁRIO
Sistema	
PRÉ CONDIÇÕES	PÓS CONDIÇÕES

Carregar arquivo compatível	Retornar os dados interpretados
FLUXO PRINCIPAL	
AÇÕES DO ATOR	AÇÕES DO SISTEMA
	carregar e interpretar os dados, apresentar ao usuário para confirmação das informações
FLUXO ALTERNATIVO I	
retornar um erro cao arquivo seja incompatível	
FLUXO ALTERNATIVO II	

A importação de dados é realizada por botões no sistema que permitem o upload de arquivos que contenham as informações correspondentes às interfaces que solicitaram, como por exemplo, ao cadastrar alunos e disciplinas, a opção de importar um arquivo csv com os dados dos alunos pode ser carregado, ou com os dados das disciplinas.

Uma etapa de verificação é necessária antes de armazenar os dados em definitivo, uma interface intermediária onde seja possível o ajuste das informações de acordo com as necessidades do formulário, por exemplo ao carregar as informações dos alunos se faz necessário a verificação ou mesmo ajuste dos campos a serem inseridos em lote.

Caso um novo formulário seja inserido ou criado por usuários gerenciadores, a interface de importação precisa comportar a quantidade de campos necessário ao novo formulário.

IDENTIFICAÇÃO DO CASO DE USO	NOME DO CASO DE USO
UC8	Gerar Relatórios
ATOR PRINCIPAL	ATOR SECUNDÁRIO
Gerenciador	
PRÉ CONDIÇÕES	PÓS CONDIÇÕES
Login ativo no sistema	
FLUXO PRINCIPAL	
AÇÕES DO ATOR	AÇÕES DO SISTEMA
Gerar relatórios de acordo com seus privilégios de acesso	
FLUXO ALTERNATIVO I	

FLUXO ALTERNATIVO II
----------------------

Relatórios incluem, informações de utilização dos alunos em relação aos formulários considerando também os períodos, possibilidade de filtrar a utilização por data dentro das limitações de permissão de acesso de cada usuário, por exemplo a coordenação e cada curso só poderá ter acesso aos dados dos alunos de seu curso. Todos os relatórios precisam considerar as datas e prazos em que foram emitidos como informação presente no corpo do mesmo.

## 2.2 BANCO DE DADOS

A necessidade de permanência dos dados é vital para tomadas de decisões frente a cenários, um histórico de atividades é fundamental do ponto de vista de tomada de decisão frente a um problema ou à necessidade de mudança, ou mesmo pela simples motivação de melhoria das atividades.

Com isso em mente, juntamente com a demanda por sistemas online, faz-se necessário a utilização de bancos de dados relacionais a fim de trazer dados de utilização e de decisões anteriores.

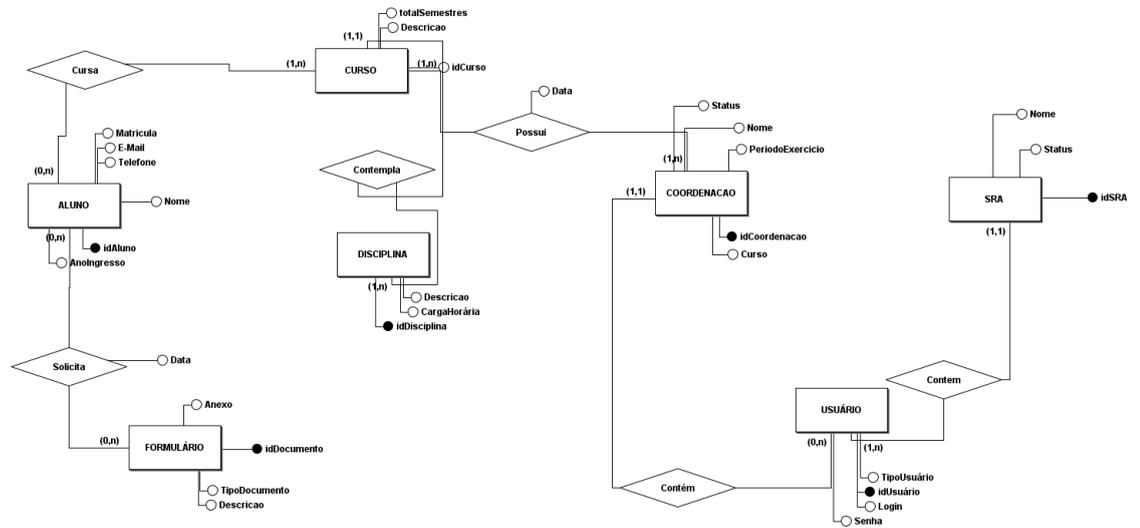
A utilização de bancos de dados relacional é tradicional em sistemas desse tipo devido a sua confiabilidade, praticidade e facilidade de manutenção, além da vasta literatura que aborda o tema, tudo isso traz segurança à implementação do sistema.

### 2.2.1 DIAGRAMA ENTIDADE RELACIONAMENTO

Esse diagrama traz um panorama da estrutura das tabelas do banco de dados, seus atributos e relacionamentos, pode descrever os tipos de dados e as cardinalidades entre os relacionamentos das entidades.

Cada entidade representa algo que precisamos caracterizar através de um conjunto de atributos no banco de dados, ou seja, uma entidade é algo do qual se deseja guardar informações que caracterizam essa entidade, como por exemplo uma pessoa, ela possui nome, altura, idade, enfim, uma pessoa pode ser identificada por uma entidade e suas aracterísticas são os atributos dessa entidade pessoa.

Figura 6: Diagrama entidade relacionamento



Fonte: Própria

## 2.2.2 MODELO LÓGICO

O modelo lógico é um modelo organizacional mais completo das relações entre as tabelas do banco de dados, é diferente do diagrama ER pois é criado pensando nas limitações e características do modelo de armazenamento utilizado para persistir os dados.

Por exemplo, uma pessoa pode se relacionar com outras entidades que podem ser atividades do dia a dia ou cargos em empresas. A diferença é que na vida real esses relacionamentos são diretos, não percebemos e não temos a necessidade de caracterizar de maneira muito detalhada o funcionamento das relações entre a pessoa e o restante, porém em um banco de dados ao representar logicamente as relações entre as entidades, precisamos considerar a organização adequada dos dados para que não haja duplicidade, perda de dados como informações ou até relações erradas entre as entidades, por isso o diagrama lógico busca produzir um diagrama que represente essas situações de maneira mais objetiva possível.

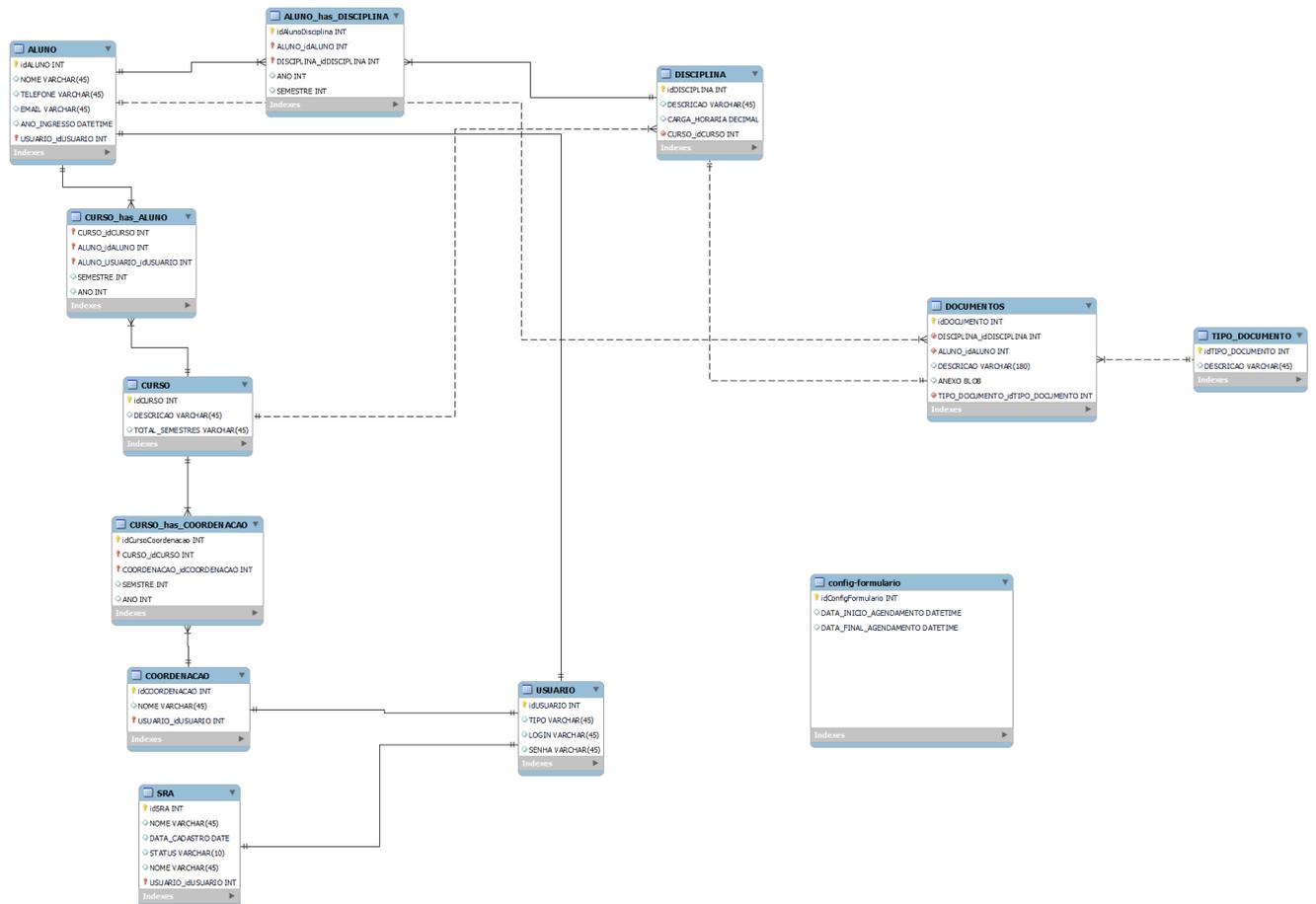
A tabela de disciplinas, por exemplo, armazena os dados relacionados à disciplina, como descrição, código de identificação do PPC, id único, semestre de oferta, além de manter um vínculo com um curso e ainda estar vinculada aos

alunos. A vinculação aos alunos acaba gerando uma tabela de apoio que tem um id próprio para cada entrada de dados, id de aluno e id da disciplina, devido à relação NxN existente, já que um aluno poderá cursar mais de uma vez a mesma disciplina e cada disciplina terá N alunos.

São armazenados os dados dos alunos como nome, ano de ingresso, matrícula, e-mail, curso, id e telefone na tabela chamada aluno, esta tem uma relação com a tabela disciplina e usuário.

Tipos de usuário são armazenados na tabela usuário, os tipos são necessários para que seja possível conceder níveis de acesso diferentes de acordo com o cargo ou papel do usuário

Figura 7: Modelo Lógico



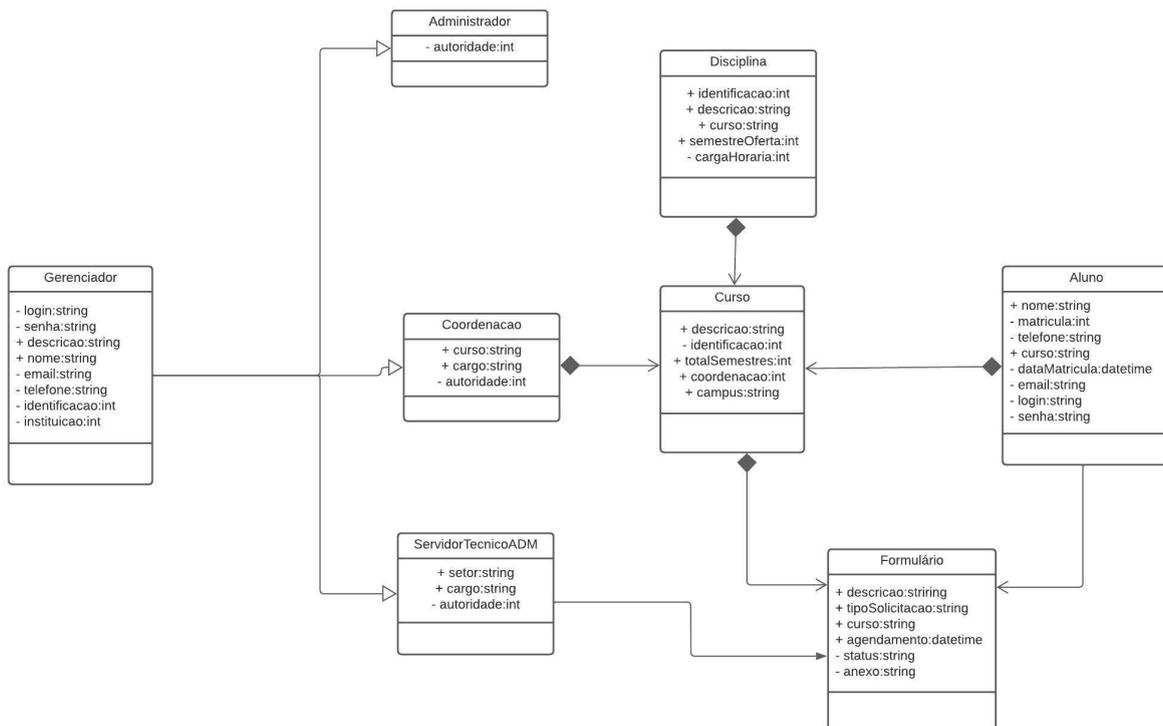
Fonte: Própria

## 2.3 DIAGRAMA DE CLASSES

Devido ao paradigma de programação orientado a objetos, o diagrama de classes, vem a facilitar o entendimento, organização e caracterização das classes

de objetos e seus atributos em um sistema. Sua construção se dá pela abstração do mundo real, dessa maneira somos capazes de perceber classes de objetos caracterizáveis, seus atributos (características) e suas funções/operações (métodos) bem como as relações entre si.

Figura 8: Diagrama de Classes

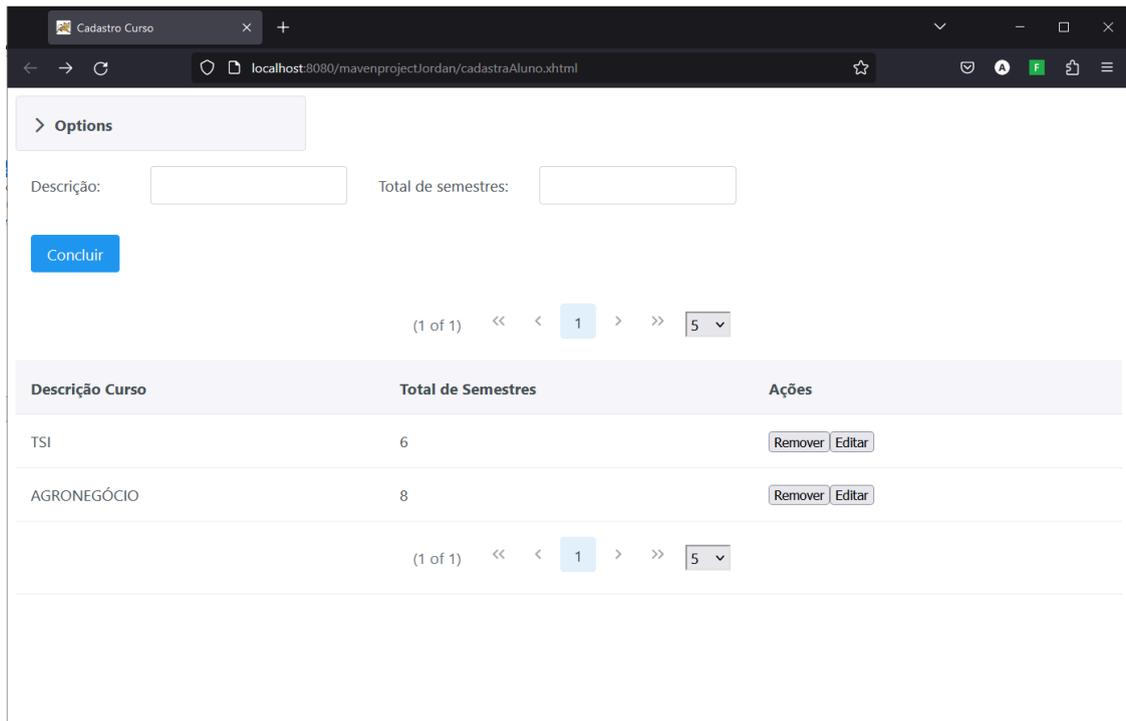


Fonte: Própria

## 2.4 TELAS DO SISTEMA

Abaixo vemos a tela de cadastro de Curso, é possível informar uma descrição do Curso bem como a quantidade de semestres do mesmo

Figura 9: Cadastro de Cursos



Fonte: própria

Na tela de Cadastro de Disciplinas é possível informar uma descrição, bem como o código utilizado no Sistema SIGAA, e vinculá-lo ao curso respectivo

Figura 10: Cadastro Disciplinas

> Options

Descrição:  Código Disciplina:

Semestre:  Carga horária:

Curso:

(1 of 1) << < > >> 5 ▾

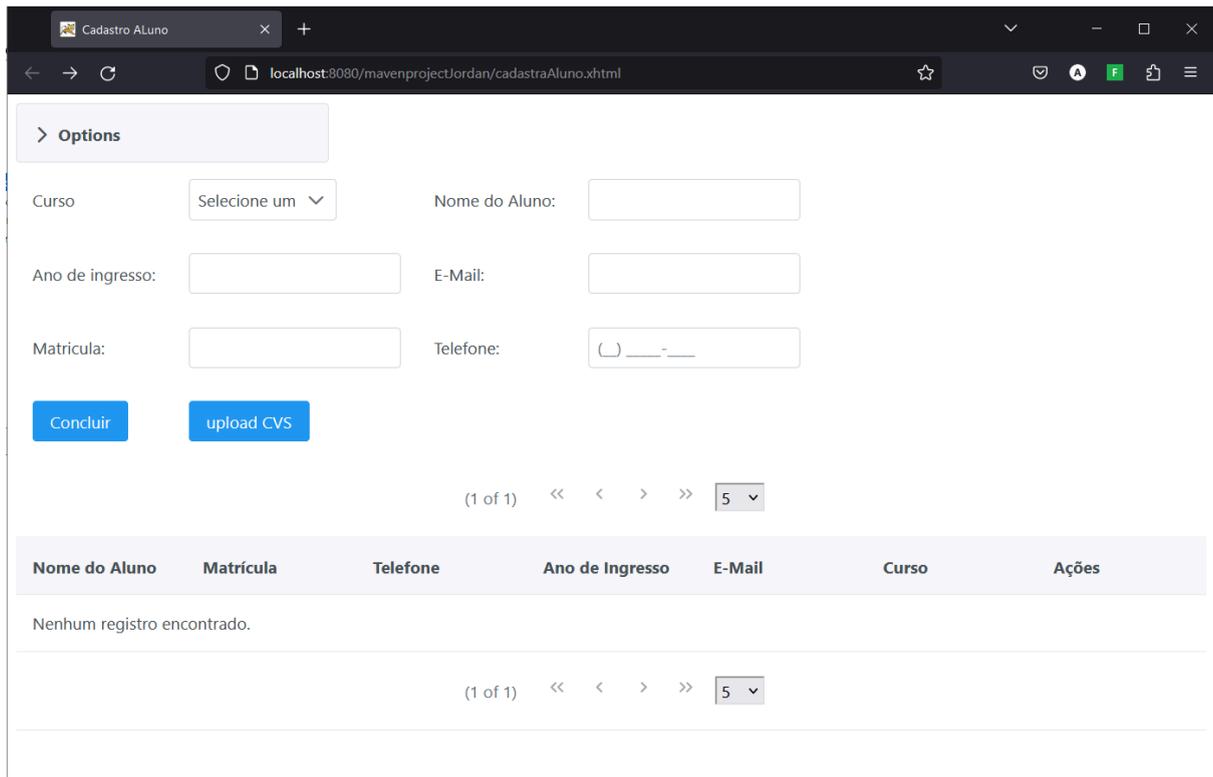
Descrição	Carga Horária	Cod PPC	Curso	Semestre	Ações
Nenhum registro encontrado.					

(1 of 1) << < > >> 5 ▾

Fonte: Própria

A seguir temos a tela de Cadastro de Alunos, onde podemos vincular ao curso, informar seu nome, matrícula, data de ingresso no curso, email e telefone.

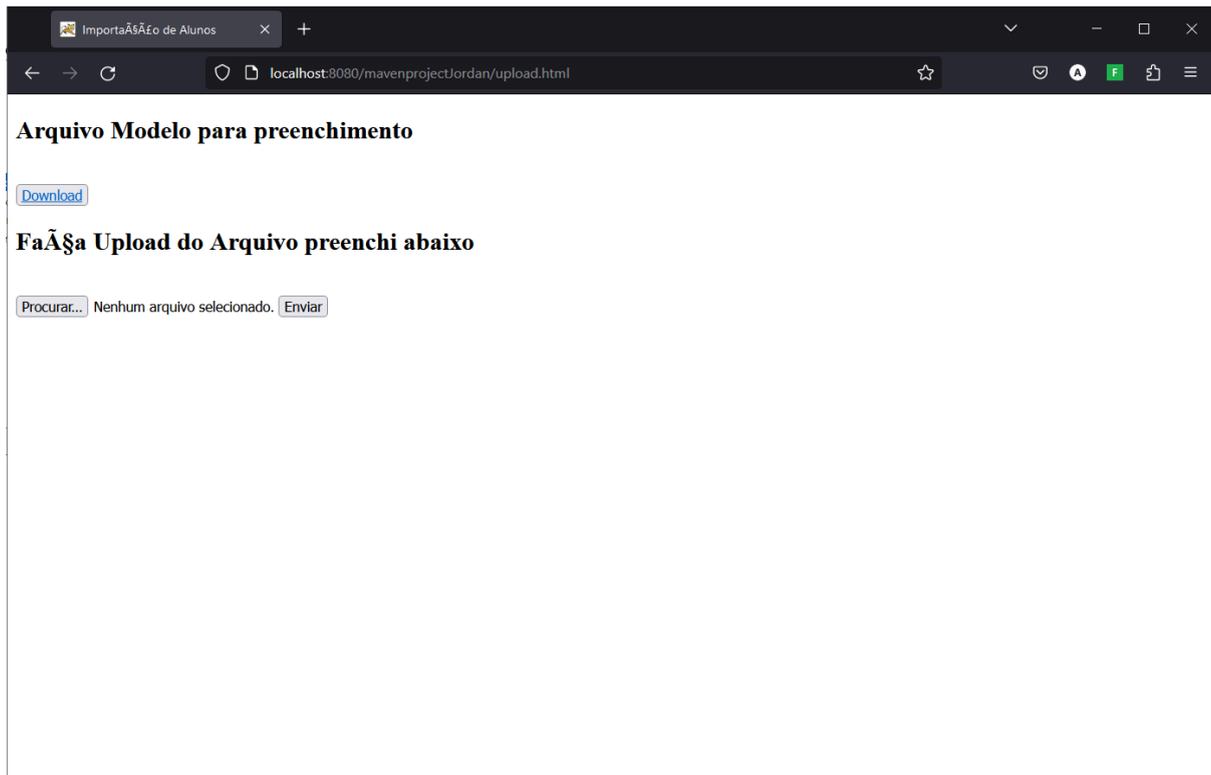
Figura 11: Cadastro de alunos



Fonte: Própria

Na tela de importação pode é possível fazer o download de um arquivo de modelo para ser preenchido, bem como o upload de um arquivo .csv preenchido.

Figura 12: Importação CSV



Fonte: Própria

## CONCLUSÃO

O sistema atualmente realiza cadastro e disciplinas, alunos bem como a importação desses dados via arquivo csv, além de permitir a criação de formulários de ajuste de matrícula, coletando as informações de curso, aluno e inclusão e exclusão de disciplinas por semestre.

Fica para um trabalho futuro a implementação dos demais tipos de requisições e a configuração de ambientes separados por tipo de usuário, bem como estatísticas de uso dos usuários que coletaram dados de localização e de plataformas utilizadas para acesso ao sistema. ainda fica aberta a possibilidade de desenvolvimento de um aplicativo para facilitar ainda mais o acesso aos dados, além de login com o google, já que a instituição tem parte de seu sistema rodando nessa plataforma.

A linguagem se mostrou desafiadora, existem inúmeras formas de realizar a mesma tarefa, porém nem sempre é possível mesclar o uso de várias diretamente, como na importação dos dados de arquivo .csv, não foi possível contornar o empecilho de ter que remover o cabeçalho do arquivo, mesmo após pesquisas e

testes com diferentes soluções. Outro desafio foi o uso de uma ideia sem utilização de frameworks, as páginas e o código em si foram todos regidos a mão, com a ajuda do orientador que tem um bom conhecimento da linguagem.

## REFERÊNCIAS

<https://www.rfc-editor.org/info/rfc4180> ACESSO EM MARÇO 2023

<https://www.w3.org/TR/xml/> ACESSO EM MARÇO 2023

<https://www.json.org/json-en.html> ACESSO EM MARÇO DE 2023

ELMASRI, R.; NAVATHE, S. B. Sistemas de banco de dados. 6. ed. São Paulo: Pearson, c2011.

What is HTTP?, w3schools, [https://www.w3schools.com/whatis/whatis\\_http.asp](https://www.w3schools.com/whatis/whatis_http.asp) acesso em março de 2023

Deitel, P. J. Deitel. Java: Como Programar, 8a. Edição. Pearson, 2010.

Introduction to JavaServer Faces 2.x, Apache Netbeans,

<https://netbeans.apache.org/tutorial/main/kb/docs/web/jsf20-intro/> acesso em 16 de março de 2023

Prime Faces Documentation, primefaces.github.io,

[https://primefaces.github.io/primefaces/11\\_0\\_0/#/](https://primefaces.github.io/primefaces/11_0_0/#/) acesso em 16 de março de 2023

Hibernate Getting Started Guide, docs.jboss.org,

[https://docs.jboss.org/hibernate/orm/6.1/quickstart/html\\_single/#preface](https://docs.jboss.org/hibernate/orm/6.1/quickstart/html_single/#preface) acesso 16 de março de 2023