**SGA - Sistema de Gerenciamento Acadêmico**

**Documento de Arquitetura de Software**

**Versão 1.2**

**Histórico da Revisão**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 25/09/2017 | 1.0 | Início do preenchimento do arquivo pelo template. | Guilherme Schults., Lúcio Flávio., Paulo Henrique, Humberto de Paula |
| 20/10/2017 | 1.1 | Versão preliminar contendo informações do tamanho, desempenho e qualidade do sistema. | Guilherme Schults., Lúcio Flávio., Paulo Henrique, Humberto de Paula |
| 13/11/2017 | 1.2 | Versão preliminar contendo informações da visão lógica, dos processos, implantação e implementação do sistema. | Guilherme Schults., Lúcio Flávio., Paulo Henrique, Humberto de Paula |
|  |  |  |  |

**Índice Analítico**

[1. INTRODUÇÃO 4](#_Toc498283962)

[1.1 Finalidade 4](#_Toc498283963)

[1.2 Escopo 4](#_Toc498283964)

[1.3 Definições, Acrônimos e Abreviações 4](#_Toc498283965)

[1.4 Referências 4](#_Toc498283966)

[1.5 Visão Geral 5](#_Toc498283967)

[2. REPRESENTAÇÃO ARQUITETURAL 5](#_Toc498283968)

[3. METAS E RESTRIÇÕES DA ARQUITETURA 6](#_Toc498283969)

[3.1 Disponibilidade 7](#_Toc498283970)

[3.2 Segurança 7](#_Toc498283971)

[3.3 Manutenibilidade 7](#_Toc498283972)

[3.4 Usabilidade 7](#_Toc498283973)

[4. VISÃO DE CASOS DE USO 7](#_Toc498283974)

[4.1. Realizações de Casos de Uso 7](#_Toc498283975)

[5. VISÃO LÓGICA 18](#_Toc498283976)

[5.1 Visão Geral 18](#_Toc498283977)

[5.1.1 Camada Modelo (Model) 18](#_Toc498283978)

[5.1.2 Camada Visão (View) 18](#_Toc498283979)

[5.1.4 Database 19](#_Toc498283980)

[5.2 PACOTES DE DESIGN SIGNIFICATIVOS DO PONTO DE VISTA DA ARQUITETURA 19](#_Toc498283981)

[6. VISÃO DE PROCESSOS 20](#_Toc498283982)

[6.1. Processo Leves: 20](#_Toc498283983)

[6.2. Processo Pesados: 20](#_Toc498283984)

[7. VISÃO DE IMPLANTAÇÃO 21](#_Toc498283985)

[8. VISÃO DA IMPLEMENTAÇÃO 22](#_Toc498283986)

[8.1 Visão Geral 22](#_Toc498283987)

[8.2 Camadas 23](#_Toc498283988)

[9. VISÃO DE DADOS 23](#_Toc498283989)

[10. TAMANHO E DESEMPENHO 23](#_Toc498283990)

[11. QUALIDADE 24](#_Toc498283991)

**Documento de Arquitetura de Software**

# INTRODUÇÃO

Este documento fornece uma visão geral e descritiva de como é a arquitetura do software Sistema de Gerenciamento Acadêmico. Este software é uma solução utilizada por instituições de ensino superior no âmbito federal, para que os alunos, professores e coordenadores possam utilizá-lo para suas atividades diárias.

## 1.1 Finalidade

Oferecer uma visão geral arquitetural abrangente do sistema, usando diversas visões arquiteturais para representar diferentes aspectos do sistema. O objetivo deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas em relação ao sistema.

## 1.2 Escopo

Definir a modelagem de dados e a estrutura de camadas do sistema, como também os fluxos e formas de comunicação entre estas camadas de forma a orientar padrões para o desenvolvimento e evolução.

## 1.3 Definições, Acrônimos e Abreviações

SGA: Sistema de Gerenciamento Acadêmico: software utilizado pelos discentes e docentes da universidade para realização de suas atividades acadêmicas diárias.

MVC: Arquitetura de software baseada em três camadas (Model, View, Controller).

API (ou JPA): Application Programming Interface para persistência de dados.

Hibernate: framework utilizado para implementação do JPA, para persistir as classes java em bancos de dados relacionais.

GUI: Graphic User Interface

## 1.4 Referências

* Slides das aulas ministradas pelo professor Gilmar.
* Hibernate: <http://hibernate.org/>
* The “4+1” view model of architecture, Philippe Kruchten, november 1995: <http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/2003/Pbk4p1.pdf>

## 1.5 Visão Geral

O documento está organizado em visões: primeiramente definindo a representação arquitetural do software, explicitando em seguida as metas e restrições da arquitetura. Em sequência temos a visão dos casos de uso, onde é ilustrado o funcionamento do software e da integração dos casos de uso com o software em si. Após isso, na visão lógica são exibidos os dados referentes ao código, mostrando as classes, métodos e os atributos mais relevantes. Temos também a visão de processos, onde verifica-se a integração do software, e a visão de implantação, que mostra a configuração física do software. Por fim, temos a visão de implementação, que é a responsável pelas camadas do software, e também a parte que define a dimensão do software e o impacto da arquitetura na qualidade do mesmo.

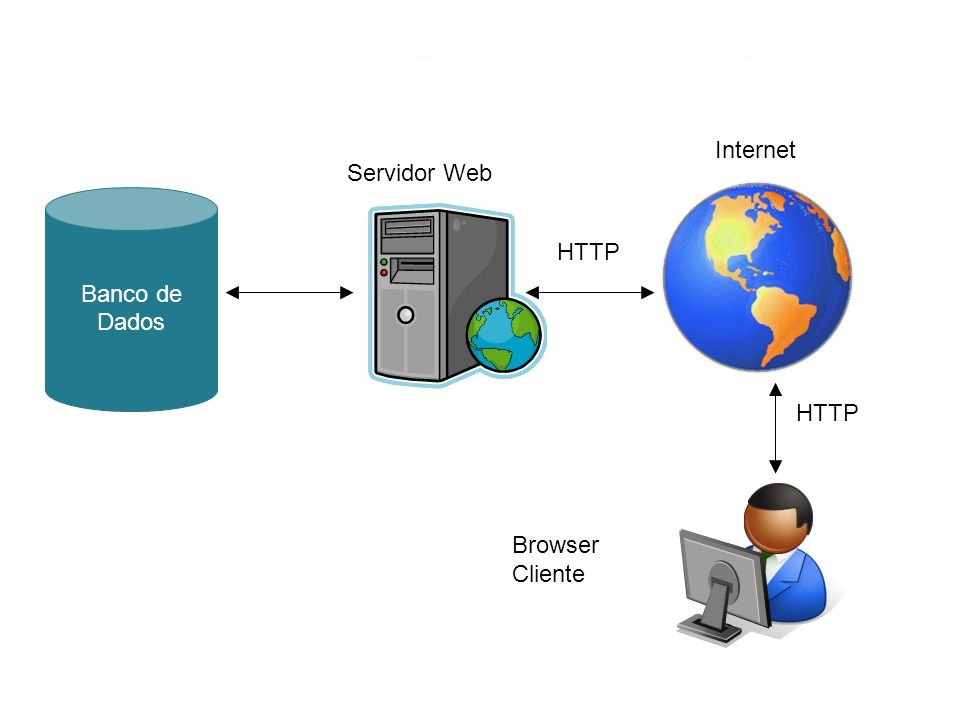
# REPRESENTAÇÃO ARQUITETURAL

A arquitetura deste software é baseada no padrão arquitetural MVC (Model-View-Controller) para deixar o sistema mais estruturado e organizado e também nas visões baseadas no modelo “4+1”. A camada Visão (View) é relacionada ao visual da aplicação, ou seja, as telas que serão exibidas ao usuário. A camada Controle (Controller) atua como intermediária entre as regras do negócio (camada Model) e a Visão, realizando o processamento de dados informados pelo usuário e repassando-os para outras camadas. Por fim, na camada Modelo (Model), encontram-se as regras de negócio, envolvendo as classes do sistema e o acesso aos dados através de validações. Para realizar o mapeamento do objeto relacional da camada View e da camada Model, utiliza-se a API Hibernate, para garantir que os dados tenham integridade e para simplificar os processos.

As visões segundo o modelo “4+1” são:

|  |  |
| --- | --- |
| Visão | Público Alvo |
| Lógica | Analistas |
| Processo | Stakeholders |
| Implementação | Programadores |
| Implantação | Stakeholders |
| Casos de Uso | Todos |

Também será utilizado o Estilo Arquitetural Cliente/Servidor, onde por intermédio da internet, o cliente realizará uma comunicação com o servidor, que então disponibilizará recursos para o cliente. Veja abaixo imagem ilustrativa deste estilo:



# METAS E RESTRIÇÕES DA ARQUITETURA

As restrições e metas estabelecidas para o projeto Sistema de Gerenciamento Acadêmico se encontram descritas abaixo. Visando à qualidade do sistema foram estabelecidos os seguintes critérios:

## 3.1 Disponibilidade

O sistema deverá estar disponível no modo 24/7, ou seja, 24 horas por dia, 7 dias por semana, para atender as mais variadas demandas quando necessário por um aluno, professor ou qualquer outro usuário.

## 3.2 Segurança

O sistema deverá ter alta segurança. Deve oferecer proteção aos seus usuários e ao ambiente em que está inserido, sendo altamente resistente a intrusões acidentais ou intencionais. Além disso, deve manter clara a segurança na persistência dos dados, pois é fundamental a integridade dos dados nessa aplicação.

## 3.3 Manutenibilidade

O sistema deverá oferecer facilidade a sua necessidade de manutenções. Dessa forma será seguido padrões durante seu desenvolvimento, de modo que facilite essas manutenções e garanta a segurança na adição de novos requisitos que podem vir a existir.

## 3.4 Usabilidade

O sistema deverá ser desenvolvido e implementado de forma responsiva, para que os usuários possam ter uma melhor experiência possível de utilização da aplicação, nos mais diversos dispositivos.

# VISÃO DE CASOS DE USO

## Realizações de Casos de Uso

CSU01 - *Manter Professores Cadastrados*

O coordenador realiza operações de manutenção dos professores cadastrados no sistema.

**FB - Fluxo Básico - Pesquisar Professor**

1. Sistema apresenta uma tela de pesquisa com uma listagem dos professores cadastrados.
2. O ator (coordenador) realiza uma busca com os dados requisitados.
3. O sistema apresenta os dados solicitados pelo ator.
4. Término do caso de uso.

**Fluxos Alternativos**

**Fluxo-A01 - Cadastro de Professor**

1. Este CSU inicia-se quando o ator solicita cadastrar um professor ao sistema.
2. O sistema apresenta uma tela de cadastro.
3. O ator informa os dados para cadastro.
4. O ator confirma a ação.
5. Sistema valida os dados informados. **[Fluxo-E03] [Fluxo-E04]**
6. O sistema persiste os dados no banco. **[Fluxo-E02]**
7. Sistema retorna para o passo 1 do fluxo básico.

**Fluxo-A02 - Alteração de Cadastro de Professor**

1. Este CSU inicia-se quando o ator solicita a alteração de um cadastro de um professor ao sistema.
2. O ator informa quais os campos serão alterados
3. O sistema valida os campos. **[Fluxo-E03] [Fluxo-E04]**
4. O ator confirma a ação.
5. O sistema persiste os dados no banco. **[Fluxo-E02]**
6. Sistema retorna para o passo 1 do fluxo básico.

**Fluxo-A03 - Exclusão de Cadastro de Professor**

1. Este CSU inicia-se quando o ator solicita a exclusão de um cadastro de um professor ao sistema.
2. O ator seleciona qual o professor terá o cadastro excluído.
3. O sistema solicita uma confirmação de exclusão ao ator.
4. O ator confirma a ação.
5. O sistema persiste a exclusão no banco. **[Fluxo-E02]**
6. Sistema retorna para o passo 1 do fluxo básico.

**Fluxos de Exceção**

**Fluxo-E01 - Dados não localizados no sistema**

1. Essa exceção ocorre quando o ator realiza uma busca, e nenhum alvo requisitado é localizado.
2. Sistema apresenta uma mensagem ao ator de que os dados não foram localizados.
3. Sistema retorna para o passo 1 do fluxo básico.

**Fluxo-E02 - Falha na Transação**

1. Essa exceção ocorre quando o sistema não consegue executar uma determinada ação.
2. Sistema realiza um rollback dos dados, desfazendo todas as alterações realizadas durante a transação.
3. Sistema retorna ao início do fluxo.

**Fluxo-E03 - Dados Obrigatórios não preenchidos**

1. Essa exceção ocorre quando o ator não informa os dados de algum campo obrigatório.
2. Sistema apresenta ao ator os campos que não foram preenchidos.
3. Sistema coloca o campo não preenchido em evidência para o ator.
4. Sistema retorna ao início do fluxo.

**Fluxo-E04 - Campos Inválidos**

1. Essa exceção ocorre quando o ator informa dados inválidos para algum campo obrigatório.
2. Sistema apresenta ao ator os campos que foram preenchidos com dados inválidos.
3. Sistema coloca o campo em evidência para o ator.
4. Sistema retorna ao início do fluxo.

CSU02 - *Gerenciar Disciplina*

O coordenador realiza operações de manutenção das disciplinas cadastrados no sistema.

**FB - Fluxo Básico - Pesquisar Disciplina no Sistema**

1. Sistema apresenta uma tela de pesquisa com uma listagem das disciplinas que estão cadastradas.
2. O ator (coordenador) realiza a busca por uma disciplina. **[Fluxo-A01]**
3. Sistema busca os dados requisitados.
4. Sistema mostra os dados solicitados. **[Fluxo-E01]**
5. Término do caso de uso.

**Fluxos Alternativos**

**Fluxo-A01 - Cadastrar Disciplina**

1. Este CSU inicia-se quando o ator solicita o cadastro de uma disciplina no sistema.
2. O ator seleciona a opção de cadastrar disciplina.
3. Sistema apresenta uma tela de cadastro.
4. Ator informa os dados nos campos de cadastro.
5. Sistema valida os dados informados pelo ator. **[Fluxo-E03] [Fluxo-E04]**
6. Sistema persiste os dados no banco. **[Fluxo-E02]**
7. Sistema retorna para o passo 1 do fluxo básico.

**Fluxo-A02 - Alteração de Disciplina**

1. Este CSU inicia-se quando o ator solicita a alteração de uma disciplina no sistema.
2. O ator seleciona a opção de alterar disciplina.
3. Sistema apresenta uma tela de alteração com os dados da disciplina cadastrada.
4. Ator informa os novos valores a serem alterados e confirma a ação.
5. Sistema valida os dados informados pelo ator. **[Fluxo-E03] [Fluxo-E04]**
6. Sistema persiste os dados no banco. **[Fluxo-E02]**
7. Sistema retorna para o passo 1 do fluxo básico.

**Fluxo-A03 - Exclusão de Disciplina**

1. Este CSU inicia-se quando o ator solicita a exclusão de um cadastro de uma disciplina do sistema.
2. O ator seleciona qual disciplina terá será excluída.
3. O sistema solicita uma confirmação de exclusão ao ator.
4. O ator confirma a ação.
5. Sistema valida os dados informados pelo ator. **[Fluxo-E03] [Fluxo-E04]**
6. Sistema persiste os dados no banco. **[Fluxo-E02]**
7. Sistema retorna para o passo 1 do fluxo básico.

**Fluxos de Exceção**

**Fluxo-E01 - Dados não localizados no sistema**

1. Essa exceção ocorre quando o ator realiza uma busca, e nenhum alvo requisitado é localizado.
2. Sistema apresenta uma mensagem ao ator de que os dados não foram localizados.
3. Sistema retorna para o passo 1 do fluxo básico.

**Fluxo-E02 - Falha na Transação**

1. Essa exceção ocorre quando o sistema não consegue executar uma determinada ação.
2. Sistema realiza um rollback dos dados, desfazendo todas as alterações realizadas durante a transação.
3. Sistema retorna ao início do fluxo.

**Fluxo-E03 - Dados Obrigatórios não preenchidos**

1. Essa exceção ocorre quando o ator não informa os dados de algum campo obrigatório.
2. Sistema apresenta ao ator os campos que não foram preenchidos.
3. Sistema coloca o campo não preenchido em evidência para o ator.
4. Sistema retorna ao início do fluxo.

**Fluxo-E04 - Campos Inválidos**

1. Essa exceção ocorre quando o ator informa dados inválidos para algum campo obrigatório.
2. Sistema apresenta ao ator os campos que foram preenchidos com dados inválidos.
3. Sistema coloca o campo em evidência para o ator.
4. Sistema retorna ao início do fluxo.

CSU03 – *Gerenciar Controle de Disciplina*

O coordenador realiza a operação de busca e exclusão da disciplina.

**FB - Fluxo Básico – Remoção da Disciplina**

1. O sistema fornece a listagem de disciplina.
2. O ator (coordenador) seleciona a disciplina que se encontra na listagem.
3. O sistema solicita a confirmação da exclusão.
4. O sistema valida a exclusão retirando a disciplina do banco de dados.
5. A disciplina é retirada da listagem de disciplina.
6. Término do caso de uso.

**Fluxos Alternativos**

**Fluxo-A01 - Cadastrar Disciplina**

1. Este CSU inicia-se quando o ator solicita o cadastro de uma disciplina no sistema.
2. O ator seleciona a opção de cadastrar disciplina.
3. Sistema apresenta uma tela de cadastro.
4. Ator informa os dados nos campos de cadastro.
5. Sistema valida os dados informados pelo ator.
6. Sistema persiste os dados no banco.
7. Sistema retorna para o passo 1 do fluxo básico.

**Fluxo-A02 - Alteração de Disciplina**

1. Este CSU inicia-se quando o ator solicita a alteração de uma disciplina no sistema.
2. O ator seleciona a opção de alterar disciplina.
3. Sistema apresenta uma tela de alteração com os dados da disciplina cadastrada.
4. Ator informa os novos valores a serem alterados e confirma a ação.
5. Sistema valida os dados informados pelo ator.
6. Sistema persiste os dados no banco. **[Fluxo-E02]**
7. Sistema retorna para o passo 1 do fluxo básico.

**Fluxos de Exceção**

**Fluxo-E01 – Disciplina não existente**

1. Essa exceção acontece quando a disciplina não é localizada.
2. O sistema não apresenta a disciplina na listagem de disciplinas cadastradas.

**Fluxo-E02 - Falha na Exclusão**

1. Essa exceção acontece quando não tem permissão de realizar a exclusão.
2. O sistema permite que apenas o usuário nível coordenador possa excluir Disciplinas.
3. Sistema solicita para que faça login com permissão nível Coordenador.

CSU04 - *Visualizar Notas*

O Aluno acessa as suas notas cadastradas no sistema.

**FB - Fluxo Básico - Listar Notas do Aluno**

1. Este CSU inicia-se quando o ator solicita visualizar suas notas.
2. Sistema apresenta uma tela com as notas do aluno.
3. O ator visualiza suas notas.
4. Término do caso de uso.

**Fluxos Alternativos**

**Fluxo-A01 - Visualizar Notas Detalhadas**

1. Este CSU inicia-se quando o ator seleciona uma disciplina.
2. Sistema apresenta uma tela com informações detalhadas sobre a nota selecionada pelo ator. **[Fluxo-E01]**
3. Sistema retorna para o passo 1 do fluxo básico.

**Fluxos de Exceção**

**Fluxo-E01 - Dados não localizados**

1. Este CSU inicia-se quando o ator seleciona uma disciplina e a mesma não apresenta informação detalhada.
2. Sistema apresenta uma informação ao ator sobre o ocorrido e comunica o professor responsável.
3. Sistema retorna para o passo 1 do fluxo básico.

CSU05 – *Gerenciar Fluxo Disciplina*

O Aluno solicita matricula em uma disciplina.

**FB - Fluxo Básico - Listar Notas do Aluno**

1. Este CSU inicia-se quando o ator solicita visualizar as disciplinas disponíveis a ele.
2. Sistema apresenta uma tela com as disciplinas disponíveis.
3. O ator seleciona uma disciplina.
4. O sistema apresenta uma tela confirmando a matrícula na disciplina selecionada.
5. Término do caso de uso.

**Fluxos Alternativos**

**Fluxo-A01 - Alterar Disciplina.**

1. Este CSU inicia-se quando o ator seleciona que deseja alterar disciplinas cadastradas.
2. Sistema apresenta uma tela com as disciplinas cadastradas.
3. O ator seleciona a disciplina que deseja excluir.
4. O sistema apresenta uma tela confirmando a exclusão da disciplina selecionada.
5. Sistema retorna para o passo 1 do fluxo básico.

**Fluxos de Exceção**

**Fluxo-E01 - Disciplina com turma cheia.**

1. Este CSU inicia-se quando o ator seleciona uma disciplina e a mesma não apresenta mais vagas.
2. Sistema apresenta uma informação ao ator sobre o ocorrido e comunica o aluno para solicitar a matrícula em outra disciplina.
3. Sistema retorna para o passo 1 do fluxo básico.

CSU06 - *Gerenciar Notas*

O professor pode cadastrar notas a alunos vinculados a sua disciplina.

**FB - Fluxo Básico - Cadastrar Nota**

1. Este CSU inicia-se quando o ator solicita cadastrar notas para alunos associados a sua disciplina.
2. Sistema apresenta uma tela com os alunos matriculados.
3. O ator seleciona um aluno.
4. O sistema apresenta uma tela para cadastro da nota.
5. O ator confirma a nota digita e retorna ao passo 1. **[Fluxo-E01]**
6. Caso não queira cadastrar mais notas o caso de uso é encerrado.

**Fluxos Alternativos**

**Fluxo-A01 - Alterar Nota Cadastrada**

1. Este CSU inicia-se quando o ator seleciona a opção alterar uma nota cadastrada.
2. Sistema apresenta uma tela com os alunos associados à sua disciplina.
3. O ator seleciona qual aluno deseja alterar a nota.
4. O sistema apresenta uma tela confirmando a nova nota digitada. **[Fluxo-E01]**
5. Sistema retorna para o passo 1 do fluxo básico.

**Fluxos de Exceção**

**Fluxo-E01 - Campos Inválidos**

1. Essa exceção ocorre quando o ator informa dados inválidos para algum campo obrigatório.
2. Sistema apresenta ao ator os campos que foram preenchidos com dados inválidos.
3. Sistema coloca o campo em evidência para o ator.
4. Sistema retorna ao início do fluxo.

*CSU07 – Gerenciar vínculo entre aluno e disciplina*

O professor pode verificar quais alunos estão cadastrados em suas disciplinas.

**FB - Fluxo Básico - Visualizar Alunos**

1. Este CSU se inicia quando o ator Professor solicita visualizar alunos cadastrados em uma disciplina.

2. Sistema apresenta uma tela com as disciplinas vinculadas ao Professor.

3. O ator seleciona a disciplina que deseja verificar.

4. O ator confirma visualizar a disciplina.

5. O sistema apresenta uma lista de alunos cadastrados.

6. Término do caso de uso.

**Fluxos Alternativos**

**Fluxo-A01 - Disciplinas vinculadas ao Professor**

1. Este CSU inicia-se quando o ator professor seleciona a opção de visualizar disciplinas vinculadas a ele.

2. O ator seleciona a opção desejada.

3. O sistema apresenta-lhe uma lista de disciplinas vinculadas a ele.

4. Término do caso de uso.

**Fluxos de Exceção**

**Fluxo-E01 – Disciplina não cadastrada**

1. Essa exceção ocorre quando o sistema não apresenta ao ator uma certa disciplina.

2. O sistema aguarda para verificação de uma disciplina válida.

3. O sistema permite o ator cadastro da disciplina. **[Fluxo-A01]**

4. Sistema retorna ao início do perfil do Professor.

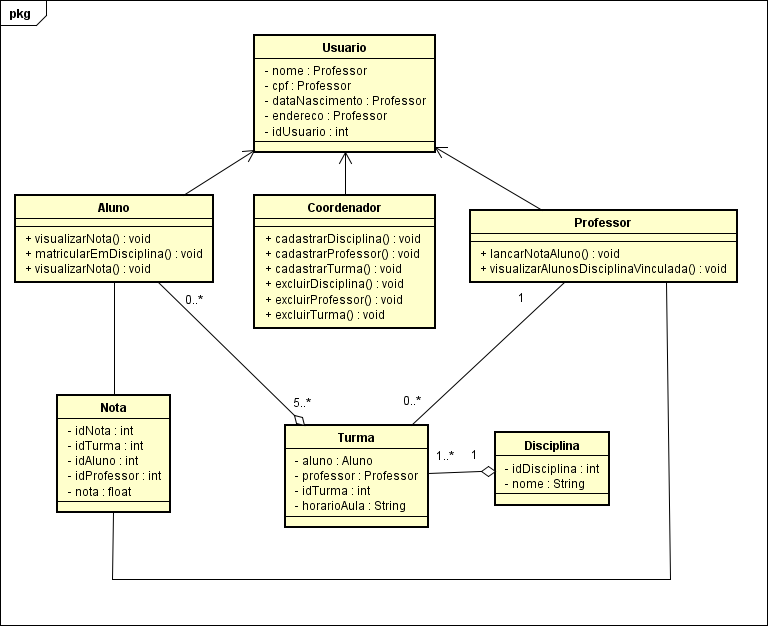
**Fluxo-E02 – Aluno não vinculado**

1. Essa exceção ocorre quando o aluno não está cadastrado na disciplina.

2. O sistema lista os alunos cadastrado.

3. O sistema permite a vinculação de um aluno a uma disciplina pelo ator Aluno.

# 5. VISÃO LÓGICA



## 5.1 Visão Geral

O sistema implementa o modelo MVC, portanto o mesmo está dividido em camadas, que são:

### **5.1.1 Camada Modelo (Model)**

Estão presentes as regras de negócio da aplicação, as validações e a persistência dos dados, além das classes que definem os objetos que são utilizados.

### **5.1.2 Camada Visão (View)**

É a camada de interface com o usuário, usada para receber a entrada de dados e apresentar os resultados solicitados (mostrando ao usuário as funcionalidades e/ou os dados armazenados no banco de dados, seja para visualização ou acréscimo de algum conteúdo). O GUI (Graphic User Interface) está implantada em CSS (com Bootstrap) e HTML5, as funcionalidades em JavaScript utilizando o framework jQuery.

5.1.3 Camada Controle (Controller)

É a camada que realiza a intermediação entre as requisições feitas pelos usuários na camada View e a camada Model.

### **5.1.4 Database**

É a camada acessada pelo controle para buscar e/ou retirar informações, e exibi-las na visualização do usuário. Executa-se também comandos do modelo através da camada de controle. O objetivo da camada de acesso ao banco de dados é manter a integridade dos dados, facilitando a comunicação com o sistema independente da sua estrutura interna.

## 5.2 PACOTES DE DESIGN SIGNIFICATIVOS DO PONTO DE VISTA DA ARQUITETURA

**Data Access Object - DAO**

O Sistema utiliza um GenericDao para evitar a utilização de códigos redundantes, garantindo alta reusabilidade e produtividade. Será responsável por encapsular o acesso aos dados organizando sua camada, ou seja, separando as regras de negócio de regras de acesso aos dados. Todas as classes de acesso ao banco de dados implementam o GenericDao, que se estende às funcionalidades específicas de cada objeto.

**Ajax**

O sistema utiliza tecnologias como JavaScript, para tornar páginas web mais interativas com o usuário.

# 6. VISÃO DE PROCESSOS

O sistema é gerenciado por meio de processos, processo leve, é a unidade básica de utilização da CPU. Consiste de um apontador de programa, um conjunto de registradores e uma espaço de pilha. Uma thread compartilha, com outras threads, a sua área de código, de dados e os recursos do sistema operacional, como arquivos abertos e sinais. Esse conjunto é denominado de tarefa (task). Um processo tradicional, ou processo pesado é uma task com uma thread. Uma task não faz nada se não possuir nenhuma thread, e uma thread só pode estar em uma task, a divisão desses processos é definida de acordo com a capacidade de influência para o sistema:

## 6.1. Processo Leves:

* Cadastrar usuário: é armazenado no banco de dados informação de usuário que posteriormente irá usufruir do sistema;
* Login de usuário: é solicitado para servidor acesso ao sistema de usuário já cadastrado;
* Inserir dados: é armazenado no banco de dados conteúdos que são inseridos no sistema;
* Visualização de dados: é solicitado para o servidor permissão para visualizar certas informações, como resposta é retornado para o usuário que solicitou;

## 6.2. Processo Pesados:

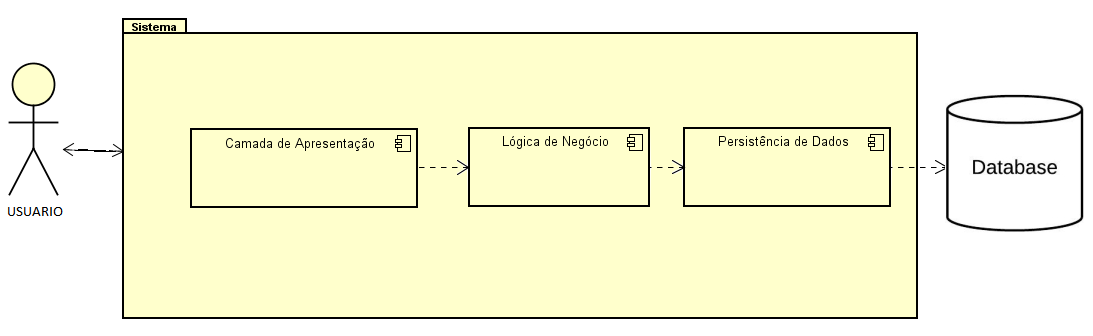
* Alteração de dados cadastrais: é solicitado para o banco de dados a substituição de certos dados que o sistema permite sua alteração;
* Exclusão de cadastro: é solicitado para o banco de dados a remoção dos dados cadastrais de um usuário do sistema;
* Exclusão de dados: é solicitado para o banco de dados a remoção de dados no sistema;
* Listagem de usuários: é solicitado para o sistema a visualização e a listagem de usuários (Alunos) vinculados a uma Disciplina;
* Listagem de disciplinas: é solicitado para o sistema a visualização e a listagem de disciplinas vinculadas a um Professor;

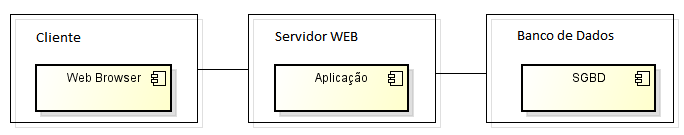
# 7. VISÃO DE IMPLANTAÇÃO

O sistema opera em nuvem, e sua implantação se dará através de um sistema cliente-servidor, onde o servidor disponibiliza o acesso web através de protocolos HTTP para qualquer computador conectado à internet. O sistema será reajustado de acordo com a demanda dos usuários, e, para isso, utilizará um serviço terceirizado como o Auto Scaling AWS da Amazon, que permite aumentar ou reduzir automaticamente e de modo dinâmico, a escalabilidade do sistema.

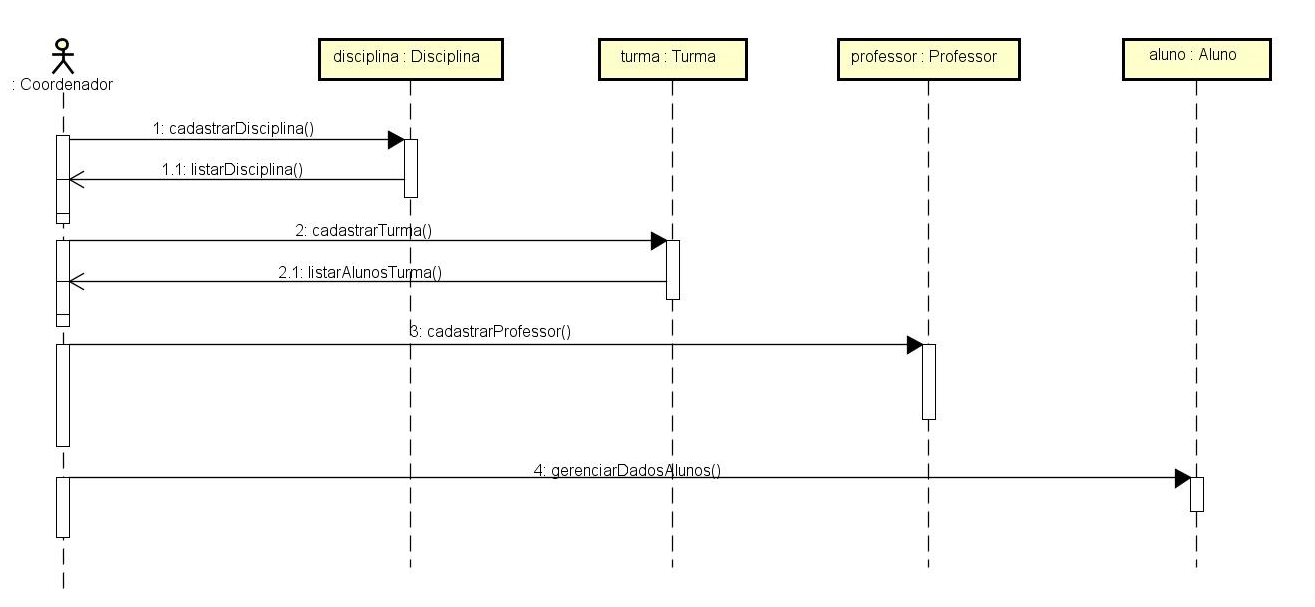
O enlace dos dados, ou seja, a ligação entre os dispositivos de comunicação, será de alta velocidade, proporcionando ao usuário uma melhor experiência de utilização do sistema. Do lado do cliente é necessário um navegador web (qualquer browser) com conexão a internet para que seja exibido às views.

A imagem abaixo ilustra o ambiente de implantação do software:

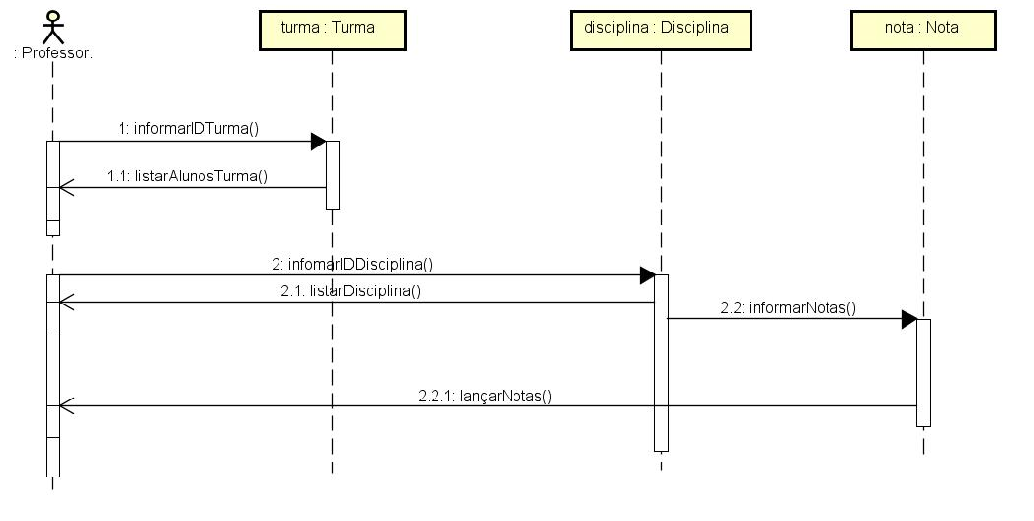


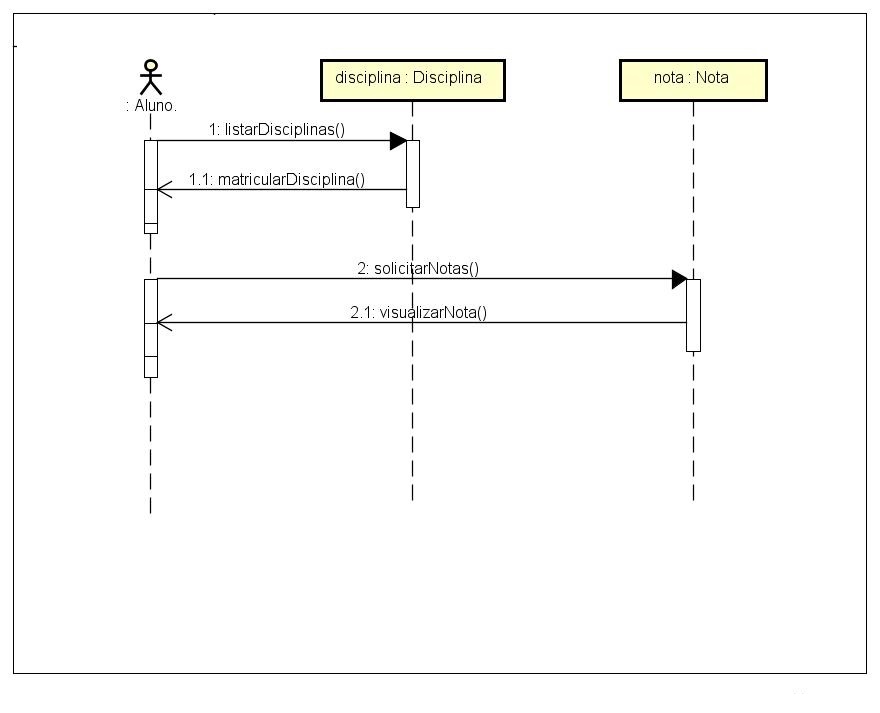


# 8. VISÃO DA IMPLEMENTAÇÃO

**

## 

**

**

## 8.1. Visão Geral

A visão de implementação mostra a distribuição física do processamento no sistema. O sistema possui três camadas principais:

* a camada de interface com o usuário
* a camada de dados
* a camada de conexão com o servidor externo

## 8.2. Camadas

As camadas estão organizadas da seguinte maneira:

Camada de interface com o usuário:

* Tela de Login do sistema
* Página Inicial
* Para cada usuário específico é redirecionado uma página inicial específica (Por exemplo, para os alunos, professores e coordenadores).

Camada de dados:

* Todos os dados dos usuários armazenados aqui nesta camada.

Camada de conexão com servidor externo:

- Conexão com a API.

# 9. VISÃO DE DADOS

*[Uma descrição da perspectiva de armazenamento de dados persistentes do sistema. Esta seção será opcional se os dados persistentes forem poucos ou inexistentes ou se a conversão entre o Modelo de Design e o Modelo de Dados for trivial.]*

# 10. TAMANHO E DESEMPENHO

* A arquitetura do sistema foi projetada para suportar simultaneamente até 2.500 usuários.
* As pesquisas feitas pelos usuários devem retornar os dados em até, no máximo, 10 segundos de latência.
* O sistema deve ser capaz de completar 85% de suas operações em até 2 minutos.
* Tempo médio de sessão de um usuário em 30 minutos.

# 11. QUALIDADE

A qualidade do sistema está em conformidade com as normas de qualidade de software propostas pela ISO/IEC 9126-1:2001, que é um padrão internacional para avaliação de software. Dentre as quais podemos citar:

|  |  |  |
| --- | --- | --- |
| **Atributos de Qualidade** | **Escolha da Decisão Arquitetural** | **Justificava para o Atributo de Qualidade e a Decisão Arquitetural** |
| Portabilidade | Aplicação Web | O sistema utiliza Aplicação Web, que pode ser acessada através dos diversos navegadores existentes no mercado, rodando em diferentes dispositivos, atingindo assim um maior número possível de usuários. |
| Manutenibilidade | Estilo Arquitetural Orientado a Objetos | Como o próprio nome diz, um paradigma de estruturas baseado na divisão de responsabilidades, classes e objetos autossuficientes, cada um contendo dados e comportamentos relevantes a seu objeto. Pode-se utilizar também técnicas de orientação a objetos como polimorfismo e herança para garantir que o software tenha alta coesão (classes bem definidas) e um baixo acoplamento (classes mais independentes). |
| Confiabilidade | Arquitetura Cliente/Servidor | O sistema deve manter em funcionamento em épocas de matrícula mesmo que apresente um tempo de resposta um pouco maior do que em épocas normais. Deverá também ter a capacidade de se manter em funcionamento mesmo em caso de falhas de software ou hardware. Por utilizar uma arquitetura cliente/servidor os serviços podem ser alugados de empresas terceirizadas as quais podem oferecer este serviço utilizando a nuvem, garantindo assim uma maior confiabilidade e maior segurança das informações. |
| Usabilidade | Design Responsivo | Justamente por ser uma aplicação web e ter a capacidade de ser acessada nos mais diversos navegadores e diferentes dispositivos, um design responsivo trará uma maior usabilidade para o usuário, de modo a ser compreensível, de fácil aprendizagem e operabilidade, atraindo o usuário de maneira simples, porém objetiva. |
| Manutenibilidade | Padrão Arquitetural MVC | O padrão MVC deixa o software mais organizado e estruturado, separando a aplicação em camadas, como: componentes de interface com o usuário, regras de negócio, lógica de aplicação e persistência de dados. Possibilitando assim uma maior organização e responsabilidades bem definidas. |
| Segurança | Arquitetura Cliente/Servidor | Através desta arquitetura os dados são armazenados em servidores na nuvem, onde possuem controles de segurança eficientes, garantindo assim maior segurança das informações. |