**<Sistema de Recomendação de Atividades>**

**Documento de Arquitetura de Software**

**Versão <1.0>**

**Histórico da Revisão**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| <08/04/2019> | <1.0> | <iniciação do documento> | <Gilvan> |
| <09/04/2019> | <1.0> | <inserção dos pontos 2,2.1,2.2,3,4,4.1,5,7,8> | <Asaph e Iara> |
|  |  |  |  |
|  |  |  |  |

**Índice Analítico**

1. Introdução 4

1.1 Finalidade 4

1.2 Escopo 4

1.3 Definições, Acrônimos e Abreviações 4

1.4 Referências 4

1.5 Visão Geral 4

2. Representação Arquitetural 4

3. Metas e Restrições da Arquitetura 4

4. Visão de Casos de Uso 5

4.1 Realizações de Casos de Uso 5

5. Visão Lógica 5

5.1 Visão Geral 5

5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura 5

6. Visão de Processos 5

7. Visão de Implantação 5

8. Visão da Implementação 5

8.1 Visão Geral 5

8.2 Camadas 6

9. Visão de Dados (opcional) 6

10. Tamanho e Desempenho 6

11. Qualidade 6

**Documento de Arquitetura de Software**

1. **Introdução**

O presente documento tem como principal objetivo mostrar a arquitetura de um sistema de recomendações de atividades para estudantes, visando mostrar suas camadas até a sua forma completa, visa também expor suas alterações e sugestões durante o processo de implementação e conclusão.

* 1. **Finalidade**

Este documento oferece 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.

Este documento tem também por objetivo expor as camadas da arquitetura do sistema de recomendação de atividades criado por alunos do curso de Licenciatura em Computação.

* 1. **Escopo**

O presente documento visa documentar as partes significativas do ponto de vista da arquitetura, como sua divisão em camadas e pacotes.

* 1. **Definições, Acrônimos e Abreviações**

CRUD - Create, Read, Update e Delete;

IDE - Ambiente Integral de Desenvolvimento;

DAO - Objeto de acesso a dados;

JSF - JavaServer Faces;

* 1. **Referências**

Não há menção de demais documentos ou relatórios externos no presente documento.

* 1. **Visão Geral**

Esse presente documento descreve detalhadamente como serão as fase de construção e desenvolvimento do projeto, dando inicial na sua introdução, para que ele foi pensado, qual sua finalidade, qual seu objetivo, que impacto isso vai acarretar na sociedade, até sua implementação, qual linguagem foi utilizada, qual framework foi adotado, qual sua arquitetura, processo para ser executado, caso de uso, desempenho e entre outras funcionalidades.

1. **Representação Arquitetural**

O SRA fará uso da arquitetura padrão MVC(model-view-controller) e usará banco de dados relacional(postgreSQL). Para a view, será usado a princípio o JSF com o framework primefaces. Para gerenciar as dependências do projeto será usado o maven, para o auxílio da configuração, o framework spring boot está em análise para uso. o versionamento do software será feito com a ferramenta git.

**2.1 Objetivos e Restrições da Arquitetura**

O projeto proposto tem como objetivo disponibilizar um sistema web onde o professor terá interação com o aluno por meio das atividades propostas, assim possibilitando maior desempenho do estudante em sala de aula.

**2.2 Arquiteturas Descartadas**

Arquitetura mobile: foi descartada visto que a equipe de desenvolvimento tem mais especialidade em sistemas web, e sendo este último útil e eficiente para atender a demanda e resolver o problema, fez-se a escolha da arquitetura web.

1. **Metas e Restrições da Arquitetura**

O sistema tem como objetivo principal garantir a segurança e a privacidade dos usuários, pensando nisso alguns dos requisitos principais são os requisitos de segurança, os usuários só poderão utilizar as funcionalidades do sistema depois que efetuar o login e o login só poderá ser efetuado por alunos cadastrados pelo professor responsável pela matéria, garantindo a descrição e confidencialidade das atividades expostas no sistema. Por questão de segurança e acessibilidade será desenvolvido um sistema em Java para Web, a arquitetura será em MVC e sua aplicação deve ser compatível com browsers de mercado ( Mozilla Firefox, Google Chrome, Opera e Safari). O sistema deve ter uma interface de fácil utilização.

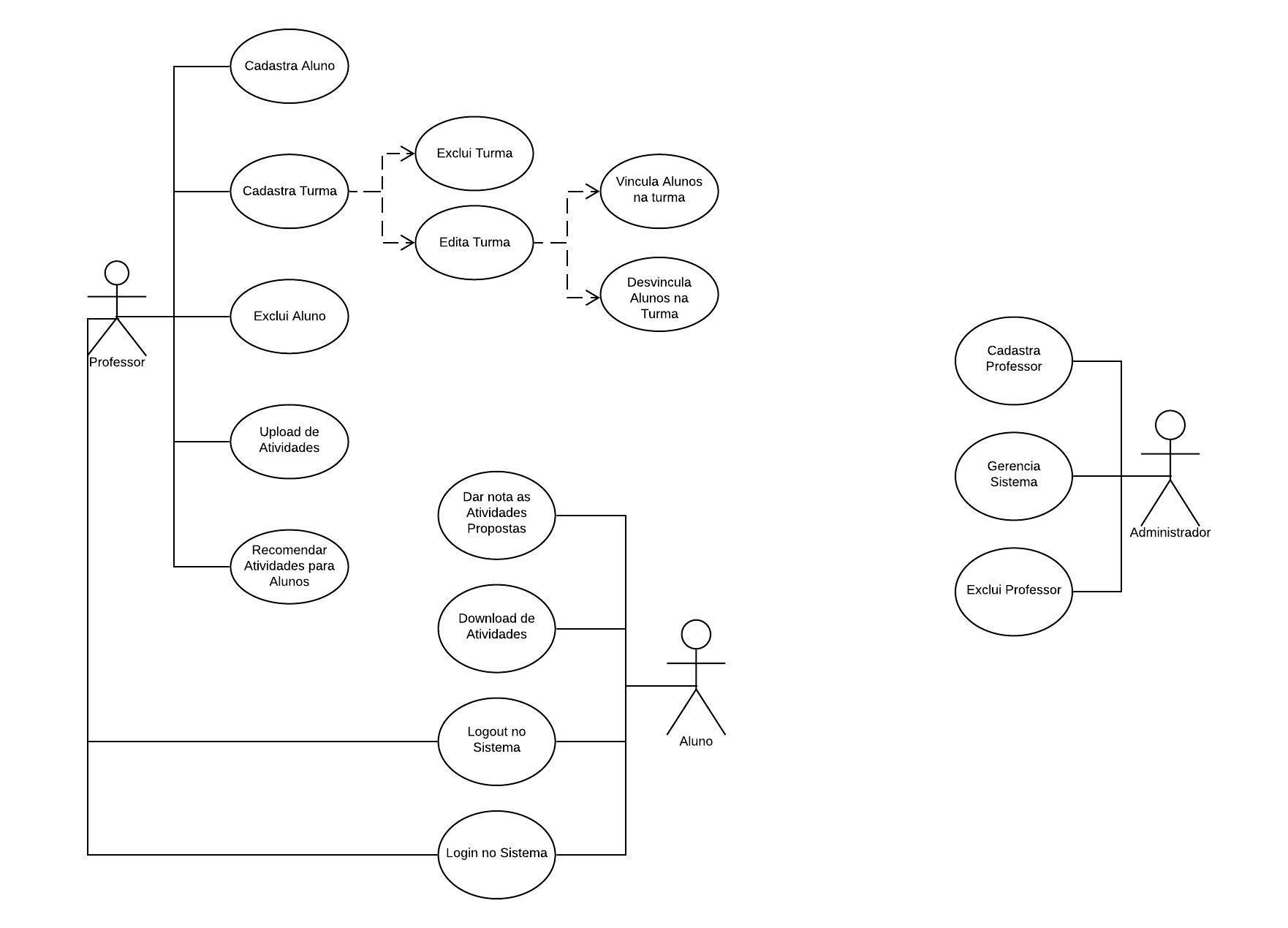
Para implementação do sistema será necessária a instalação dos seguintes softwares:

* Netbeans IDE;
* PostgreSQL;
* Git;

1. **Visão de Casos de Uso**

Login no Sistema: permite que o usuário tenha acesso às funcionalidades do sistema de acordo com o seu perfil, peça fundamental na comunicação entre usuário e sistema, já que as funcionalidades só serão liberadas depois do login.  
Logout no Sistema: O usuário pode sair da sua conta sem o risco de outra pessoa que não tenha sua senha ter acesso ao sistema.   
Cadastrar Professor: permite que o administrador do sistema cadastre adequadamente um professor, o administrador do sistema tem a total responsabilidade em liberar o sistema para o usuário professor.  
Cadastrar Aluno: permite que o professor cadastre adequadamente o aluno.  
Cadastrar Turma: permite que o usuário professor cadastre uma nova turma.  
Upload de Atividade: permite que o usuário professor coloque o arquivo ou o link da atividade.  
Download de Atividade: permite que o usuário aluno faça o download da atividade que o professor colocou no sistema.

* 1. **Realizações de Casos de Uso**



1. **Visão Lógica**

Entidades: As entidades básicas e principais do sistema são, administrador que poderá fazer o cadastro dos professores, além de gerenciar o sistema como um todo. Professor, que precisará ter os atributos nome e formação não nulos. Aluno, será cadastrado no sistema pelo professor que terá os atributos nome e turma como não nulos. Avaliação, que terá os atributos observação, título e nota. E turma, com nome, ano, observação e disciplina.

Controller: cada entidade terá o seu controller específico.

Modelo: terá um DAO genérico para fazer o CRUD das entidades, esta por sua vez terá uma interface com os métodos do CRUD e demais métodos comuns a todas as entidades e uma classe para a implementação genérica. Além disso, terá interfaces dos DAO’s específicos e suas respectivas classes de implementação que estenderão a classe de implementação do DAO genérico. Também terá todas as validações de negócio e tratamento das entradas de usuário.

view: Caso persista na utilização do framework PrimeFaces, as páginas serão em .xhtml.

* 1. **Visão Geral**

O sistema está sendo desenvolvido em mvc. Então consiste em três camadas: View, Controller e Model, sendo este último subdividido em dois pacotes: models e repositório. tem ainda o pacote de test que é onde ficará as classes de testes unitários. Um pacote entidades será criado para melhor separação das demais classes e assim facilitar a manutenção.

* 1. **Pacotes de Design Significativos do Ponto de Vista da Arquitetura**

1. **Visão de Processos**

1. **Visão de Implantação**

Para execução do software, por se tratar de um sistema web, os processos serão gerados no servidor e alguns pelo Browser do cliente. Não sendo portanto necessário uma configuração elevada de hardware da máquina que executar´o sistema. Podendo ser executado em simples computadores de escritórios até computadores gamers.

1. **Visão da Implementação**

**Camada de Apresentação**

Essa camada conterá todas as interfaces visuais, na qual interagirá diretamente com o usuário do sistema. contém a página Web, e as referências adequadas para Designer. *Observações: Deve-se se evitar qualquer tipo de desenvolvimento de código em java ou linguagem equivalente nesta camada.*

**Camada de Negócio**

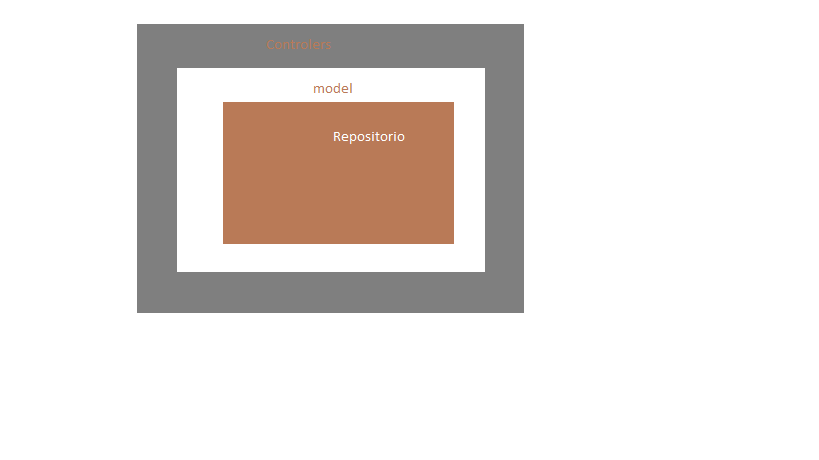
Essa camada conterá todas as interações necessárias para estruturação do negocio do cliente. *Padrão de Projeto utilizado (Facade).* Essa camada conterá todas as interações de serviço, para integrar remotamente com qualquer tipo de serviço que também possui a tecnologia para consumir WebService.

**Camada de Persistência**

Essa camada conterá todas as interações necessárias para integrar o banco de dados com os objetos de negócio, e separar todo tipo de SQL ou linguagens padrões de banco de dados, da camada de negócio. *Padrão de Projeto utilizado (Singleton, Data Access Object).*

* 1. **Visão Geral**

* 1. **Camadas**

****

1. **Tamanho e Desempenho**

O sistema é de pequeno porte, visto que será usada para suprir a necessidade de uma comunidade específica. Entretanto, não significa que esse sistema não possa se estender a demais comunidades escolares.

1. **Qualidade**

A arquitetura do software contribui para diversos fatores, uma vez que será um sistema web e por padrão a arquitetura mvc, subdivide em o sistema para facilidade de validações, requisições e persistência dos dados no BD. Também permite que em conjunto com demais padrões seja possível sua extensibilidade com mais facilidade de manutenção.