

Academicci

Documento de Arquitetura de Software

Versão 0.005

Histórico de Revisão

Data	Versão	Descrição	Autor
28/09/2017	0.000	Criação do Documento	Thaís Tavares
02/03/2018	0.001	Adequação ao Template	Raphael Guedes
03/03/2018	0.002	Inserção de informações relativas ao escopo do produto	Raphael Guedes
06/03/2018	0.003	Inserção de informações relativas ao escopo do produto	Thaís Tavares
13/08/2018	0.004	Revisão de Documentação	Thaís Tavares
19/10/17	0.000	Criação do documento	Thaís Tavares
09/11/17	0.001	Inserção diagrama de caso de uso e modelo MVC	Thaís Tavares
03/03/18	0.002	Adequação de template	Raphael Guedes
24/04/2018	0.003	Refinamento do documento: - Visão de Casos de Uso; - Visão de Lógica; - Visão de Implementação; - Visão de Dados	Raphael Guedes
29/09/2018	0.004	- Diagrama de componentes - Visão geral - Tamanho e Desempenho - Qualidade	Winnie Oliveira
08/12/2018	0.005	Alteração da estrutura do documento de arquitetura com inserção de visão de sistema.	Winnie Oliveira

Documento de Arquitetura

1. Visão

O objetivo da visão do sistema é coletar, analisar e definir necessidades e recursos de alto nível do Projeto Academicci. Ele está focalizado nos recursos necessários aos investidores e usuários de destino e por que essas necessidades existem. Os detalhes de como o projeto Academicci atende essas necessidades são explicados ao longo do documento.

1.1. Referências

- [Documento de Glossário disponível no Github;](#)
- Plano de Projeto

2. Posicionamento

2.1. Declaração do Problema

No cenário acadêmico, os alunos deparam-se com assuntos diversos, matérias que podem ou não ter visto e métodos de ensinamentos diversificados. Assim, é comum que alguns encontrem dificuldades ao longo do ciclo de aprendizado, pois o tempo dedicado pelos docentes não é suficiente para ajudá-los. Para solucionar este déficit, foi desenvolvido o projeto de monitoria acadêmica, aonde alunos se colocam a disposição para auxiliar os demais alunos. Porém, atualmente todo o procedimento de gerenciamento deste projeto é realizado de forma manual. Pensando neste pressuposto, o sistema Academicci visa o gerenciamento de monitorias para auxiliar o acadêmico na realização da monitoria seja de forma presencial ou a distância.

Este sistema advém da necessidade de controle e regulamentação do processo de seleção, acompanhamento e emissão de resultados das monitorias que poderão levar a certificação. ACADEMICCI visa melhorar o gerenciamento de modo a automatizar as funcionalidades de seleção, certificação dos monitores e acompanhamento de atividades do monitor no seu dia-a-dia, o sistema permitirá que o aluno consiga interagir em tempo real e de forma direta com seu aluno-orientador ou com o professor responsável pela disciplina a qual o aluno estiver matriculado. Outra funcionalidade fundamental em que o sistema disponibilizará é o controle de horas de acesso para que o aluno-orientador possa receber o certificado ao final do semestre, se o mesmo atingir a quantidade de horas definida em edital de monitoria.

O Problema	<ul style="list-style-type: none"> • Administração e Gestão de Monitorias realizada de forma manual
Afeta	<ul style="list-style-type: none"> • Coordenação de Monitoria • Acadêmicos do Curso de Engenharia de Computação
O Impacto do Problema é	<ul style="list-style-type: none"> • Comunicação Efetiva entre Aluno e Monitor ou Aluno e Professor • Grande demanda de trabalho atual devido a falta de uma ferramenta de padronização • Flexibilidade de atendimento e lançamento de informações
Uma Solução Bem-Sucedida Seria	<ul style="list-style-type: none"> • Auxiliar no planejamento, desenvolvimento e organização estrutural da monitoria. • Melhorar o fluxo de inscrição [de QUÊ? Ou de QUEM?], aulas, certificação e verificação das aulas executadas ao longo do semestre. • Ferramenta de interação em tempo real entre docentes, discentes e participantes da monitoria. • Ferramenta que promove ambiente para levantamento de dúvidas e questionamentos para discussão

2.2. Declaração do Produto (Elevator Statement)

Para alunos, monitores e professores universitários que necessitam de uma plataforma de integração para as monitorias. O Academicci é uma aplicação baseada em web, que integra os aspectos da monitoria em um ambiente virtual, possibilitando comunicação, controle e divulgação de conteúdo, gerando facilidade no acesso e comunicação entre alunos e mentores.

Diferentemente do controle de chamadas e conteúdos em planilhas e papéis, comunicação informal e distanciamento do docente, o produto busca facilitar a interação entre monitores, professores e alunos, oferecendo gestão do processo de monitoria, possibilitando compartilhar conhecimento de forma real e efetiva e disponibilizando uma plataforma de estudos sanando os problemas gerados atualmente. Desta forma, o produto visa automatizar os processos de inscrição, gestão e emissão de certificados dos alunos-monitores.

3. Descrição dos Stakeholders

3.1. Resumo dos Stakeholders

Nome	Descrição	Responsabilidades
Coordenador do Departamento de Monitoria (Renata D. Braga)	Docente responsável por coordenar o processo de seleção, aprovação dos monitores às disciplinas e certificação desses.	- Principal stakeholder no levantamento de requisitos do projeto

4. Visão Geral do Produto

4.1. Escopo do Produto

O Academicci é um sistema para gerenciamento de monitorias realizadas pelos próprios alunos (discentes), seja de forma presencial, seja à distância. Este sistema advém da necessidade de controle e regulamentação do processo de seleção, acompanhamento e emissão de resultados das monitorias que poderão levar à certificação. O Academicci visa melhorar o gerenciamento de modo a automatizar as funcionalidades de seleção, gestão e certificação dos monitores, no sistema haverá a possibilidade de interação real e direta entre o Aluno-Orientado ou com o Professor para tirar dúvidas constantes com relação a disciplina em questão.

4.2. Funções do Produto

As Features a serem desenvolvidas neste período, serão: Autenticação de Usuário, Cadastro de Usuário e de Disciplinas, Mural Interativo e Publicações

Funcionalidade	Prioridade
Autenticação	ESSENCIAL
Cadastro Usuários / Disciplinas	ESSENCIAL
Mural Interativo	IMPORTANTE
Publicações	IMPORTANTE

4.3. Restrições e Premissas do Produto

Descrição	Tipo
Todos os usuários deverão estar devidamente matriculados em um curso da instituição que ofereça monitoria	Restrição
Para que o aluno seja monitor ele deverá passar pela aprovação direta de um docente da disciplina em questão	Restrição
O Curso deverá disponibilizar estrutura de rede adequada para acesso ao sistema por parte dos discente e docentes	Premissas

5. Requisitos Não-Funcionais

Requisito	Prioridade	Tipo
Manutenabilidade	IMPORTANTE	
Portabilidade	ESSENCIAL	
Usabilidade	IMPORTANTE	

6. Arquitetura

6.1. Finalidade

Abranger a visão arquitetural do sistema, usando diversas visões de arquitetura para representar diferentes aspectos do sistema. Ele pretende capturar e transmitir as decisões arquiteturas significativas que foram tomadas em relação ao sistema.

6.2. Escopo

Este documento é destinado a toda equipe de desenvolvimento e serve como guia para a visão arquitetural de todo o sistema, assim como dos requisitos de implementação do sistema.

6.3. Referências

- Disciplina de Arquitetura de Software OpenUP v 1.0. Disponível em: <http://www.hectordufau.com.br/openup/index.htm>
- Documento de Glossário. Disponível em: https://github.com/Fabrica-de-Software/Academicci_II/wiki/Documento-de-Gloss%C3%A1rio
- Documento de Plano de Projeto

6.4. Visão Geral

São apresentados neste documento diferentes visões arquiteturais de como o sistema deve se comportar em diferentes processos, como deve ser implantado e implementado e restrições de desempenho e qualidade.

7. Representação da Arquitetura

O projeto Academicci usará a arquitetura MVC, que divide o sistema em três módulos: Visão, responsável pela interação com o usuário. Controller, responsável pela comunicação entre a camada de persistência e as requisições do usuário. Model, a camada de persistência de dados, que interage diretamente com o bando de dados.

- Visão de Caso de Uso

Apresenta as funcionalidades arquiteturais importantes e os usuários do sistema.

- Visão Lógica

Descreve classes e sua organização e apresenta o padrão de arquitetura que deverá ser utilizado para desenvolvimento do sistema.

- Visão de Processos

Mostra o padrão de comportamento do sistema diante de diferentes ações do usuário.

- Visão de Implantação

- Descreve a estrutura do ambiente onde o software será instalado.

- Visão de Implementação

○ Ilustra a distribuição do processo em um conjunto de nós do sistema, incluindo a distribuição física de processos e threads.

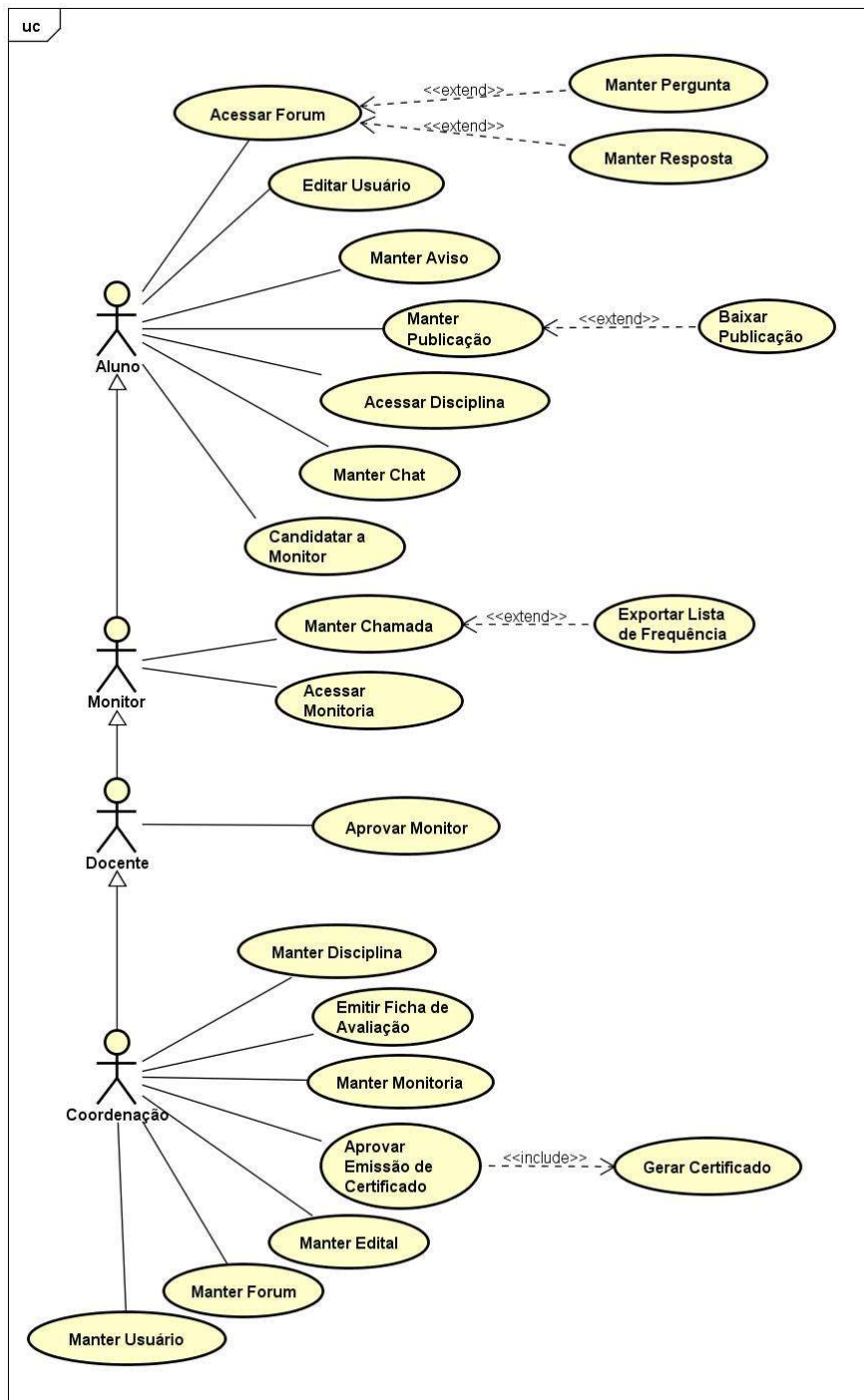
8. Metas e Restrições da Arquitetura

O sistema será desenvolvido em plataforma *web* de modo responsivo, para que seja possível sua utilização em dispositivos mobile através de um navegador web.

O projeto será implementado usando a linguagem de programação Java EE. Contará com frameworks de desenvolvimento back-end, front-end e de banco de dados. O framework back-end será o Java Server Faces (JSF), versão 2.2.12. O framework front-end, será o Primefaces, versão 6.2. E o framework de mapeamento objeto relacional será o Hibernate 4.3.11.

O software deverá manter as características de segurança, desempenho, portabilidade e confiabilidade para que atenda aos requisitos de qualidade do usuário.

9. Visão de Casos de Uso



9.1. Realizações dos Casos de Uso

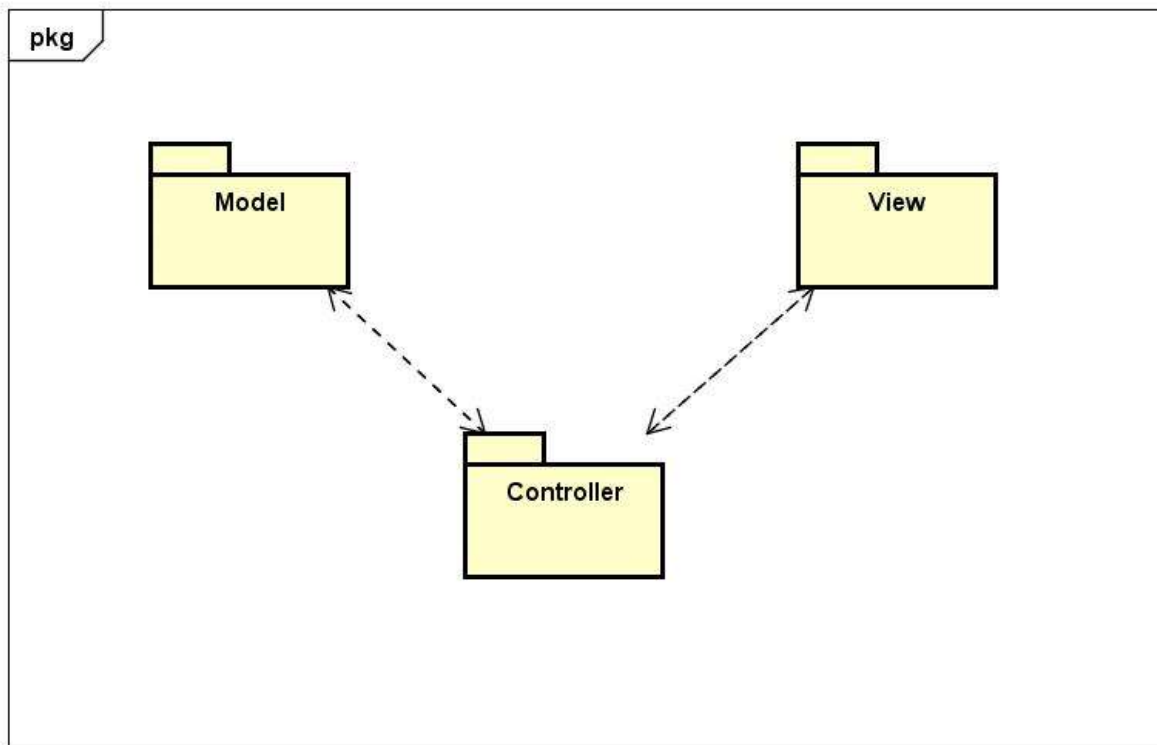
Caso de Uso	Descrição
Acessar Forum	Visualizar as publicações do Forum
Manter Pergunta	Criar, alterar ou excluir perguntas no Forum
Manter Resposta	Criar, alterar ou excluir respostas no Forum
Editar Usuário	Alterar as informações do usuário
Manter Aviso	Criar, alterar, excluir, listar, expirar um aviso
Manter Publicação	Criar, alterar, excluir, buscar uma publicação
Baixar Publicação	Fazer o download das publicações postadas
Acessar Disciplinas	Acessar as informações das disciplinas matriculadas
Manter Chat	Iniciar ou excluir chat
Candidatar a Monitor	Candidatar a vaga de monitor lançada pela coordenação
Manter Chamada	Criar, alterar, listar ou excluir lista de frequência
Exportar Lista de Frequência	Exportar as listas de frequências criadas
Acessar Monitoria	Acessar os dados da monitoria cadastrada
Aprovar Monitor	Aprovar o candidato a monitor para determinada monitoria
Manter Disciplina	Criar, alterar, listar, ativar ou desativar uma disciplina
Emitir Ficha de Avaliação	Emitir a ficha de avaliação dos monitores
Manter Monitoria	Criar, alterar, excluir, ativar ou desativar monitoria
Aprovar Emissão de Certificados	Aprovar a emissão de certificados para os monitores
Gerar Certificados	Gera os certificados aprovados dos monitores
Manter Edital	Criar, editar ou Excluir edital de monitoria
Manter Fórum	Iniciar, alterar, ativar ou desativar um fórum
Manter Usuário	Criar, alterar, ativar ou desativar um usuário

10.2. Pacotes de Design Significativos do Ponto de Vista da Arquitetura

[Para cada pacote significativo, inclua uma subseção com o respectivo nome, uma breve descrição e um diagrama com todos os pacotes e classes significativos nele contidos.

Para cada classe significativa no pacote, inclua o respectivo nome, uma breve descrição e, opcionalmente, uma descrição de algumas das suas principais responsabilidades, operações e atributos.]

11. Visão de Implementação



11.1. Visão Geral

Seguindo o padrão arquitetural MVC, o sistema está dividido em 3 camadas: Model, View e Controller.

11.2. Camadas

11.2.1. View

A camada view é responsável pela interação do usuário com o sistema.

11.2.2. Controller

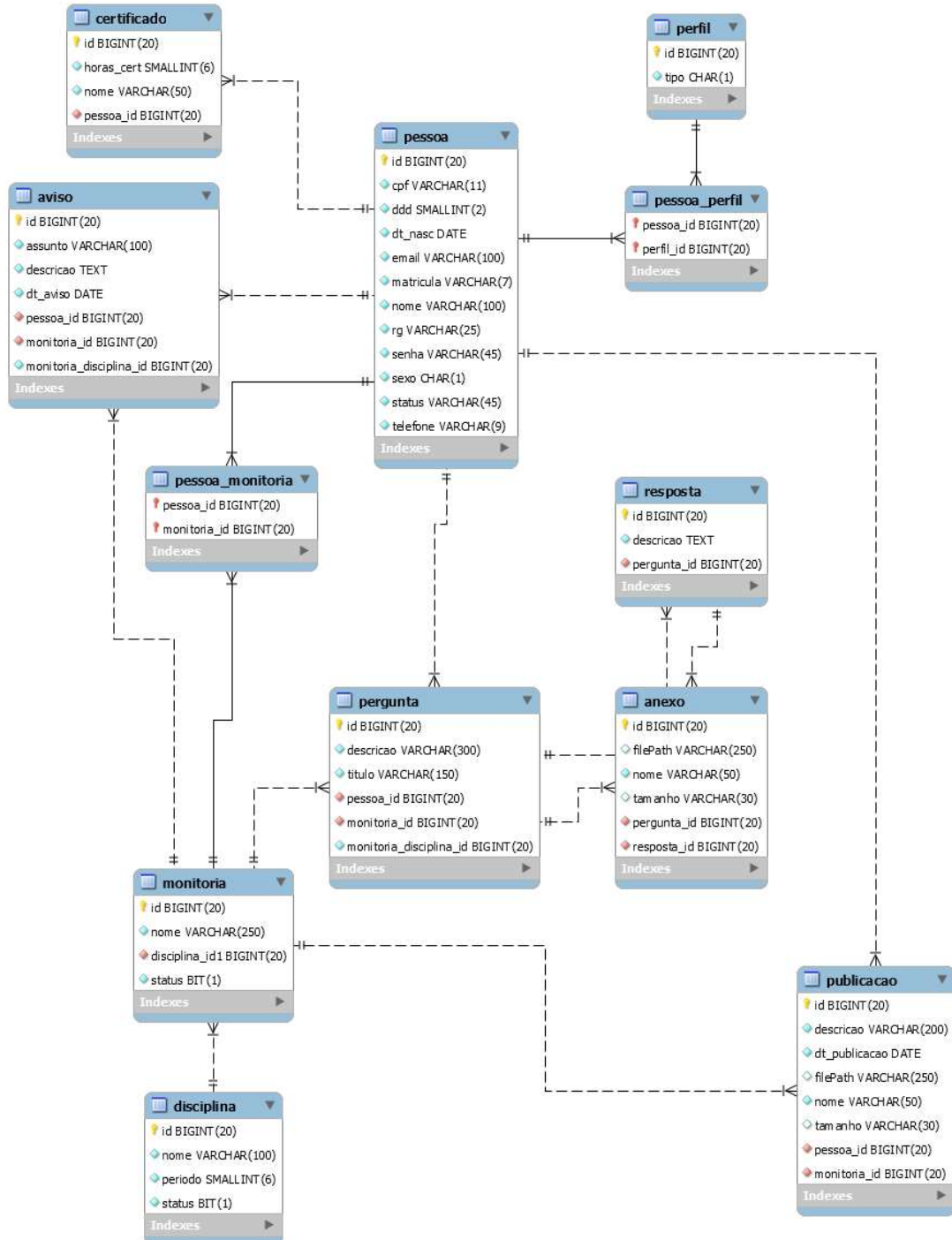
A camada controller é responsável por comunicar as requisições do usuário ao sistema.

11.2.3. Model

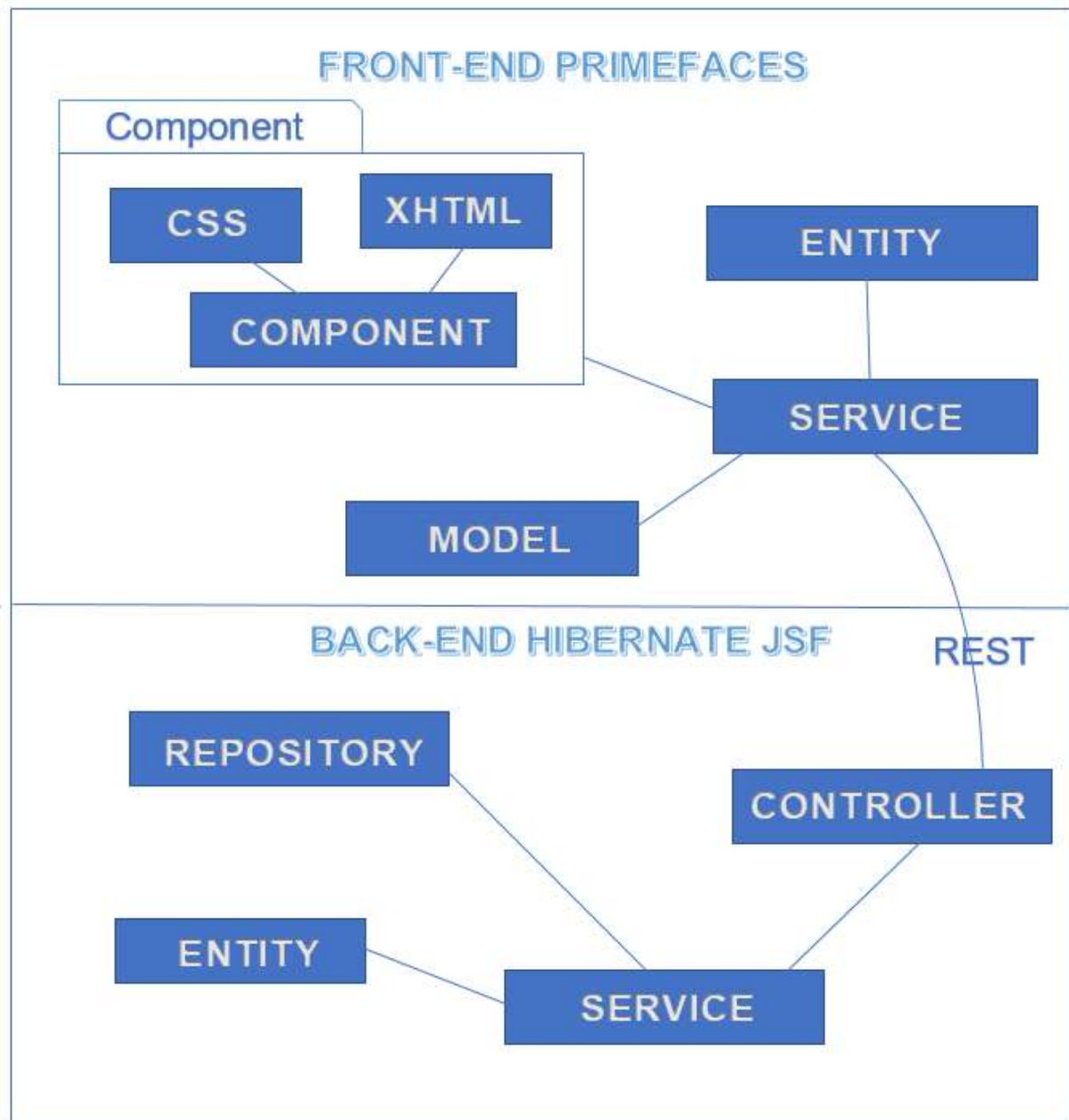
A camada model é responsável pelo armazenamento das classes no banco de dados.

12. Visão de Dados

12.1. Modelo Entidade Relacionamento – MER



13. Visão de Componentes



14. Tamanho e Desempenho

14.1. Volume

Enumerar os itens relativos ao volume de acesso aos recursos da aplicação:

- Número estimado de usuários:
- Número estimado de acessos diários:
- Tempo de sessão de um usuário:

15. Qualidade

Enumerar os itens de qualidade de software significativos para a aplicação:

Item	Descrição	Solução
Confiabilidade, Disponibilidade	O sistema deve ser funcional na ausência de conexão com a internet	Utilizar APIs em conjunto com o MYSQL para que se tenha um banco de dados em tempo real e de sincronização offline
Portabilidade	O sistema deve ser executado em um maior número possível de versões de browser	Utilizar pacotes de compatibilidade para a maioria das versões dos sistemas operacionais
Segurança	O sistema deve possuir um mecanismo de autenticação de usuários	Utilizar API de autenticação