

# SAPDI

## SISTEMA DE APOIO AO PDI

---

### Escopo e Histórias de Usuário

#### Equipe:

Leandro Ribeiro de Souza

Vínicus Assunção Marins

Vitor Augusto de Souza Rego

**Orientador(es): Danilo Codeco Carvalho e Bruno Vani**

**Cliente(s): Danilo Codeco Carvalho e Bruno Vani**

## ESCOPO DO SISTEMA

**Nome:** Sistema de Apoio ao Plano de Desenvolvimento Institucional

**Sigla:** SAPDI

O SAPDI tem como objetivo facilitar o desenvolvimento do Plano de Desenvolvimento Institucional (PDI) ao permitir o gerenciamento de informações relacionadas a reuniões de definição do PDI e a simulação de cenários de carga horária dos professores do Campus de Presidente Epitácio. O sistema possui um único nível de acesso que dá acesso a todas as funcionalidades, autenticação de servidores, redefinição de senha e proteção de rota. Além disso, o SAPDI permite o cadastro e gerenciamento de reuniões, servidores, áreas de lotação, áreas de concurso, cursos, disciplinas e servidores que serão utilizados durante simulações de cenários relativos ao Campus de Presidente Epitácio. O sistema não gera o relatório completo do PDI, mas facilita o seu desenvolvimento ao fornecer um controle de anotações das reuniões e a capacidade de simular cenários de carga horária no campus. Devido ao alto nível de instrução dos usuários, o sistema terá um baixo nível de validações, permitindo-lhes uma ampla liberdade para executar tarefas. O objetivo final do sistema é ajudar o Campus Presidente Epitácio fornecendo recursos e ferramentas para auxiliar na gestão do mesmo, e principalmente conseguir ajudar os membros da comissão do PDI de acordo com as necessidades apontadas por eles.

# HISTÓRIAS DE USUÁRIO

**Observação:** 1 dia equivale a 6 horas de trabalho.

## História 1:

"Como Usuário, eu quero me autenticar no sistema para realizar as ações que ele permite"

Detalhes adicionais: ---.

Critérios de aceitação: Deve ser possível realizar a autenticação no sistema.

Nível de complexidade e justificativa: Médio pois é necessário construir todo o fluxo de autenticação.

Estimativa de tempo: 1 dia.

## História 2:

"Como Usuário, eu quero ter a autonomia de redefinir minha senha caso eu esqueça a mesma"

Detalhes adicionais: Deve ser enviado um código de redefinição para o E-Mail.

Critérios de aceitação: A nova senha deve ser salva no banco mediante as validações (código de redefinição válido e nova senha atendendo aos critérios de segurança).

Nível de complexidade e justificativa: Médio pois é necessário construir todo o fluxo de autenticação.

Estimativa de tempo: 1 dia.

## História 3:

"Como Usuário, eu quero ter a autonomia de alterar meus dados cadastrais caso eu julgue necessário"

Detalhes adicionais: Deve ser enviado um código de redefinição para o E-Mail.

Critérios de aceitação: A nova senha deve ser salva no banco mediante as validações (código de redefinição válido e nova senha atendendo aos critérios de segurança).

Nível de complexidade e justificativa: Médio pois é necessário construir todo o fluxo de autenticação.

Estimativa de tempo: 1 dia.

## História 4:

"Como Usuário, eu quero manter as informações sobre os servidores do campus para anexá-los aos comentários de uma reunião."

Detalhes adicionais: Os servidores podem fazer comentários durante a reunião, um usuário do sistema pode registrar o autor e o comentário realizado por ele.

Critérios de aceitação: Deve ser possível realizar o CRUD de servidores.

Nível de complexidade e justificativa: Díficil pois é necessário criar todos os cenários do CRUD e listagem.

Estimativa de tempo: 1 dia.

**História 5:**

"Como Usuário, eu quero manter informações das reuniões para ter fácil consulta posteriormente."

Detalhes adicionais: Deve ser possível realizar o CRUD de servidores.

Critérios de aceitação: Deve ser permitido realizar o CRUD de reuniões.

Nível de complexidade e justificativa: Difícil pois o cadastro é extenso e é necessário criar todos os cenários do CRUD e listagem.

Estimativa de tempo: 2 dias.

**História 6:**

"Como Usuário, eu quero manter informações de cursos para usá-las nas simulações."

Detalhes adicionais: Deve ser possível realizar o CRUD de cursos.

Critérios de aceitação: Deve ser possível realizar o CRUD de Cursos.

Nível de complexidade e justificativa: Difícil pois é necessário criar todos os cenários do CRUD e listagem.

Estimativa de tempo: 1 dia.

**História 7:**

"Como Usuário, eu quero manter informações de disciplinas para usá-las nas simulações."

Detalhes adicionais: Deve ser possível realizar o CRUD de disciplinas.

Critérios de aceitação: Deve ser possível realizar o CRUD de disciplinas.

Nível de complexidade e justificativa: Difícil pois é necessário criar todos os cenários do CRUD e listagem.

Estimativa de tempo: 1 dia.

**História 8:**

"Como Usuário, eu quero manter informações de professores para usá-las nas simulações."

Detalhes adicionais: Deve ser possível realizar o CRUD de Professores. Não é necessário que o professor seja real.

Critérios de aceitação: Deve ser possível realizar o CRUD de professores.

Nível de complexidade e justificativa: Difícil pois é necessário criar todos os cenários do CRUD e listagem.

Estimativa de tempo: 1 dia.

**História 9:**

"Como Usuário, eu quero poder anexar arquivos em uma reunião para poder consultar posteriormente."

Detalhes adicionais: ---.

Critérios de aceitação: Deve ser possível adicionar e remover arquivos de uma reunião.

Nível de complexidade e justificativa: Muito Difícil pois é necessário persistir arquivos no armazenamento do sistema e integrar com o banco.

Estimativa de tempo: SEM PREVISÃO.

Observações: Necessário todos os integrantes pois é algo complexo.

**História 10:**

"Como Usuário, eu quero visualizar a distribuição de professores por área para compreender o panorama geral de professores do campus."

Detalhes adicionais: ---.

Critérios de aceitação: O gráfico deve ser legível e possuir escala.

Nível de complexidade e justificativa: Médio pois é necessário implementar o módulo de gráficos do PrimeFaces.

Estimativa de tempo: 1 dia.

**História 11:**

"Como Usuário, eu quero poder consultar o resumo de áreas para melhor compreender a situação do campus."

Detalhes adicionais: ---.

Critérios de aceitação: Tabela de exibição com o cenário editado.

Nível de complexidade e justificativa: Muito Difícil, pois é necessário criar os cálculos envolvidos e adaptá-los aos parâmetros do sistema.

Estimativa de tempo: 5 dias.

Observação: Necessário todos os integrantes pois é necessário muito debate, teste e validação.

**História 12:**

"Como Usuário, eu quero poder consultar o resumo de áreas por meio de uma tabela com boa visualização e recursos visuais indicativos para melhor compreender a situação do campus."

Detalhes adicionais: ---.

Critérios de aceitação: Tabela conforme o protótipo.

Nível de complexidade e justificativa: Médio pois é necessário tratar os dados da tabela.

Estimativa de tempo: 1 dia.

Observações: Essa história é posterior a história 11, pois é na história 11 onde os dados da tabela serão processados.

**História 13:**

"Como Usuário, eu quero poder gerar um relatório de ATA da reunião."

Detalhes adicionais: ---.

Critérios de aceitação: Seguir o template padrão para documentação de ATAs.

Nível de complexidade e justificativa: Médio pois é necessário recriar o template.

Estimativa de tempo: 1 dia.

**História 14:**

"Como Usuário, eu quero que a reunião tenha uma lista de presentes e ausentes."

Detalhes adicionais: ---.

Critérios de aceitação: --.

Nível de complexidade e justificativa: Médio pois é necessário estruturar o banco de dados e criar os relacionamentos adequados.

Estimativa de tempo: 1 dia.

## REQUISITOS NÃO FUNCIONAIS

**Confiabilidade:** O Sistema de Apoio ao Planejamento e Desenvolvimento Institucional (SAPDI) requer operação contínua, funcionando 24 horas por dia, sete dias por semana, com uma tolerância extremamente baixa de apenas 1% para falhas.

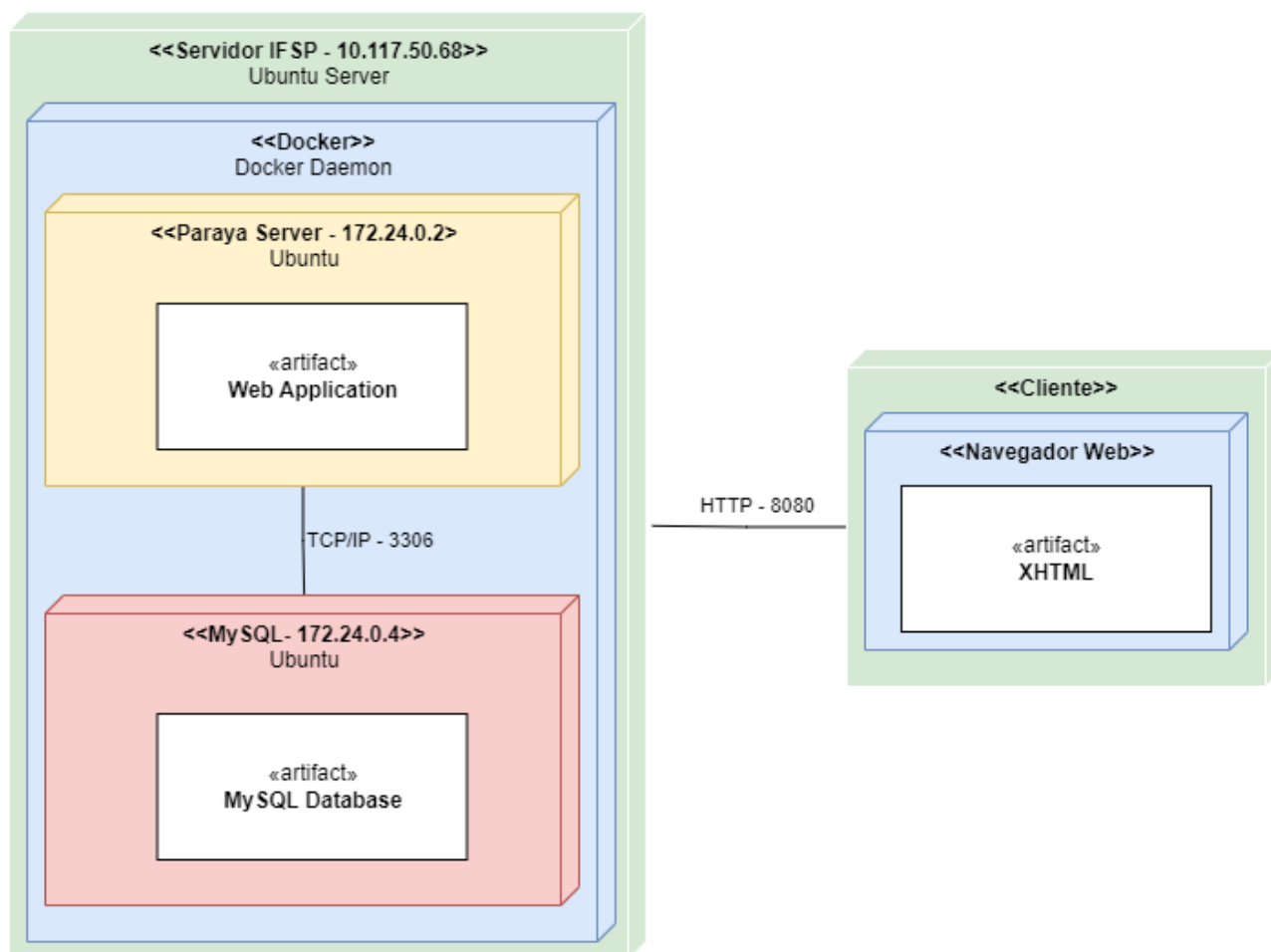
**Portabilidade:** O sistema deve ser compatível com os principais navegadores, incluindo o Mozilla Firefox na versão 98.0.2 ou anterior, bem como o Google Chrome na versão 102.0.5005.63 ou anterior.

**Implementação:** O sistema será implementado empregando a linguagem de programação Java, com base na plataforma Java EE 7. Ele será suportado por um banco de dados MySQL 8.0, hospedado em um servidor Payara Server 6, e utilizará o framework PrimeFaces 11 para a interface do usuário, além disso utilizará a tecnologia Docker para criação do ambiente virtual para hospedagem.

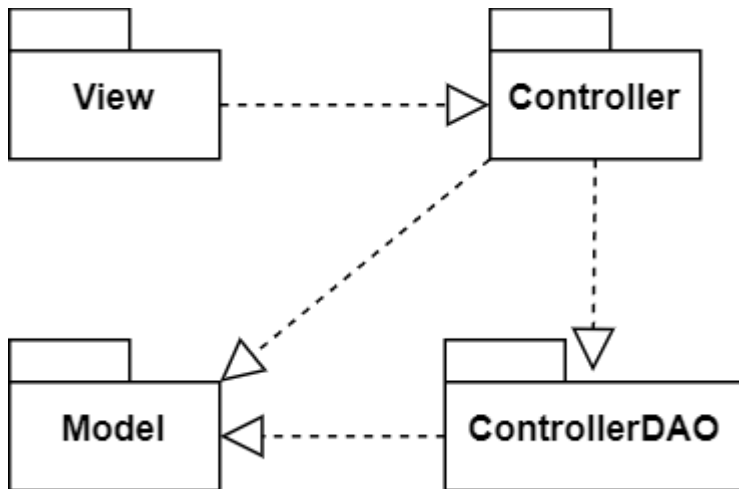
**Usabilidade:** O sistema será projetado para operar em um ambiente web e apresentará um design responsivo, mantendo uma aparência semelhante ao sistema já em uso pela Instituição para garantir uma experiência de usuário consistente e facilitada. Durante o ciclo de desenvolvimento, serão conduzidos testes de usabilidade em cada etapa para validar esse requisito.

**Desempenho:** O desempenho do sistema será projetado com foco na garantia de uma experiência de usuário aceitável, assegurando um funcionamento eficiente e sem grandes problemas. Isso significa que os usuários podem confiar que o sistema oferecerá um desempenho consistente e eficaz, tornando-o uma ferramenta valiosa para suas necessidades diárias, sem preocupações com problemas significativos.

# ARQUITETURA DE IMPLANTAÇÃO DO SISTEMA



## ARQUITETURA LÓGICA DO SISTEMA



**Model:** O Model é a representação dos dados do sistema. Ele é responsável por obter, armazenar e atualizar as informações de um objeto.

**View:** A View é a interface do usuário. Ela exibe os dados ao usuário e também coleta as entradas do usuário. A View é composta por páginas JSF, HTML e CSS.

**Controller:** O Controller atua como um intermediário entre o Model e a View. Ele processa todas as solicitações do usuário, interage com o Model e ControllerDAO para obter ou alterar dados e atualiza a View de acordo.

**ControllerDAO (Data Access Object):** O DAO é uma interface entre o Model e o banco de dados. Ele contém métodos específicos para buscar, salvar, atualizar e excluir dados no banco de dados. Isso abstrai os detalhes do banco de dados do Model.

Em resumo, em uma solicitação típica, o Controller recebe a solicitação do usuário, interage com o Model através do ControllerDAO conforme necessário, e então envia os dados para a View para serem exibidos ao usuário.





# PROTÓTIPO NAVEGÁVEL

Protótipo