

CENTRO UNIVERSITÁRIO DE ANÁPOLIS – UniEVANGÉLICA
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

**DESENVOLVIMENTO DE PROTÓTIPO DE UM SOFTWARE PARA GESTÃO DE
MONITORIA USANDO UM PROCESSO ÁGIL**

ALEXANDRE RANGEL PARREIRA RIBEIRO
WANDERSON INÁCIO DOS SANTOS

Anápolis
2018/2

ALEXANDRE RANGEL PARREIRA RIBEIRO
WANDERSON INÁCIO DOS SANTOS

**DESENVOLVIMENTO DE PROTÓTIPO DE UM SOFTWARE PARA GESTÃO DE
MONITORIA USANDO UM PROCESSO ÁGIL**

Trabalho de Conclusão de Curso II apresentado como requisito parcial para a obtenção de grau do curso de Bacharelado em Engenharia de Computação do Centro Universitário de Anápolis – UniEVANGÉLICA.

Orientador(a): Prof.^a Dr.^a Renata Dutra Braga.

Anápolis

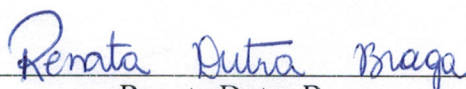
2018/2

ALEXANDRE RANGEL PARREIRA RIBEIRO
WANDERSON INACIO DOS SANTOS


**DESENVOLVIMENTO DE PROTÓTIPO DE UM SOFTWARE PARA A
GESTÃO DE MONITORIA USANDO UM PROCESSO ÁGIL**

Trabalho de Conclusão de Curso II apresentado como requisito parcial para a obtenção de grau do curso de Bacharelado em Engenharia de Computação do Centro Universitário de Anápolis – UniEVANGÉLICA.

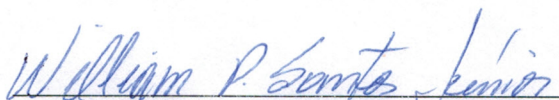
Aprovado(a) pela banca examinadora em 13 de Dezembro de 2018, composta por:



Renata Dutra Braga
Presidente da Banca



Kleber Silvestre Diogo
Prof(a). Convidado(a)



William Pereira dos Santos Júnior
Prof(a). Convidado(a)

Resumo

Introdução: Durante a graduação, alguns alunos possuem dificuldades em aprender certos conteúdos e, assim, a monitoria é uma estratégia para contribuir com a melhoria desse aprendizado, como também para reduzir o índice de evasão e reprovação em disciplinas com maior nível de dificuldade, especialmente àquelas do componente específico do curso.

Objetivo: Desenvolver um *software* de monitoria para a gestão e execução da mesma, nos Bacharelados em Computação do Centro Universitário de Anápolis.

Metodologia: Um estudo qualitativo, do tipo exploratório foi realizado com desenvolvimento de um estudo de caso. Na oportunidade, reuniões periódicas para obtenção de requisitos foram realizadas, visando obter informações sobre o processo da monitoria acadêmica. Esse estudo foi estruturado em três etapas: (a) Identificação dos requisitos e proposta de uma arquitetura capaz de garantir a qualidade e as boas práticas durante o desenvolvimento do *software*; (b) Aplicar um processo ágil para o desenvolvimento e testes do *software* em questão; e, (c) Assegurar a qualidade do protótipo desenvolvido, colocando-o em execução.

Resultados: As reuniões realizadas com a Coordenação de Monitoria dos Bacharelados em Computação permitiram identificar uma lista de requisitos que deveriam ser atendidos. Por meio do uso das etapas, atividades e boas práticas do XP (*Extreme Programming*), um protótipo de *software* foi construído durante a execução de 7 *sprints*.

Conclusão: Durante o desenvolvimento do estudo de caso a principal dificuldade foi aprender as novas tecnologias adotadas no projeto e adequá-lo, pois, ninguém da equipe tinha um conhecimento aprofundado em desenvolvimento. O uso das boas práticas do XP facilitou o conhecimento destas técnicas auxiliando tanto no desenvolvimento quanto na documentação. Assim, foi concluído que a metodologia ágil XP foi de bom proveito para que o *software* fosse bem construído com o auxílio das boas práticas adotadas.

Palavras-chave: Monitoria Acadêmica. Processo Ágil. Desenvolvimento de Software Ágil. Programação Extrema.

Abstract

Introduction: During graduation, some students have difficulties in learning certain contents and, therefore, monitoring is a strategy to contribute to the improvement of this learning, as well as to reduce the rate of avoidance and disapproval in subjects with a higher level of difficulty, especially to those of the specific component of the course.

Objective: To develop monitoring software for the management and execution of the same, in the Baccalaureate in Computing of the University Center of Anápolis.

Methodology: A qualitative exploratory study was carried out with the development of a case study. On occasion, periodic meetings to obtain requirements were made, aiming to obtain information about the process of academic monitoring. This study was structured in three stages: (a) Identification of requirements and proposal of an architecture capable of guaranteeing quality and good practices during software development; (b) Apply an agile process for the development and testing of software in question; and, (c) Ensure the quality of the developed product by putting it into execution.

Results: The meetings held with the Coordination of Monitoring of the Bachelor of Computing allowed to identify a list of requirements that should be met. Through the use of XP (Extreme Programming) steps, activities and good practices, a software prototype was built during the execution of 7 sprints.

Conclusion: During the development of the case study the main difficulty was to learn the new technologies adopted in the project and adapt it, therefore, no one of the team had an in-depth knowledge in development. The use of XP's best practices facilitated the knowledge of these techniques, aiding in both development and documentation. Thus, it was concluded that the agile XP methodology was of good benefit so that the software was well constructed with the aid of the good practices adopted.

Keywords: Academic Monitoring. Agile process. Agile Software Development. Extreme Programming.

LISTA DE ILUSTRAÇÕES

Figura 1: Modelo de histórias de usuários e critérios de aceitação.....	15
Figura 2: Estrutura do Processo XP.....	17
Figura 3: Diagrama de Caso de Uso.....	27
Figura 4: Protótipos utilizados no início do projeto.....	29
Figura 5: Modelo lógico do MER.....	30
Figura 6: Visão dos componentes.....	31
Figura 7: Código fonte da tela de login.....	32
Figura 8: Estruturação do código do <i>software</i>	33
Figura 9: Práticas do XP.....	34
Figura 10: Ferramenta Github.....	36
Figura 11: Ferramenta Bitbucket.....	37
Figura 12: Trello – Gerenciamento de tarefas.....	37
Figura 13: Tela inicial do software Academicci.....	38
Figura 14: Tela de Usuários do <i>software</i> Academicci.....	39
Figura 15: Código fonte da Tela de Usuários.....	40
Figura 16: Porcentagem de código testado.....	43
Figura 17: Exemplo de Teste de Unitário.....	43
Figura 18: Exemplo de Teste de Integração.....	44
Figura 19: Resultados dos Testes feitos no Cadastrar Disciplinas.....	44

LISTA DE TABELAS

Tabela 1: Requisito Não-Funcional.....	28
Tabela 2: <i>Sprints</i> Trabalhadas.....	36

LISTA DE SIGLAS E ABREVIACÕES

CSS	<i>Cascading Style Sheets</i>
FTT	Fábrica de Tecnologias <i>Turing</i>
HTML	<i>Hypertext Markup Language</i> (Linguagem de Marcação de Hipertexto)
IDE	<i>Integrated Development Environment</i> (Ambiente Integral de Desenvolvimento)
JSF	<i>JavaServer Faces</i>
KIS	<i>Keep It Simple</i> (Mantenha Simples)
MER	Modelo Entidade-Relacionamento
MVC	<i>Model-View-Controller</i> (Modelo-Visão-Controlador)
OO	Orientado a Objeto
UML	<i>Unified Modeling Language</i> (Linguagem de Modelagem Unificada)
XP	<i>Extreme Programming</i> (Programação Extrema)

SUMÁRIO

1.	INTRODUÇÃO.....	11
2.	OBJETIVOS.....	13
2.1.	Objetivo Geral.....	13
2.2.	Objetivos Específicos	13
3.	METODOLOGIA.....	14
3.1.	Identificação dos requisitos, e propor uma arquitetura que garanta a qualidade e as boas práticas de desenvolvimento.....	15
3.2.	Aplicação do processo ágil para o desenvolvimento e testes do <i>software</i> em questão.....	16
3.3.	Assegurar a qualidade do protótipo desenvolvido.....	18
4.	REFERENCIAL TEÓRICA.....	20
4.1.	Monitoria.....	20
4.2.	Processo Ágil.....	20
4.2.1.	Programação Extrema.....	20
4.2.1.1.	Práticas da Programação Extrema.....	21
Feedback.....		21
Comunicação		21
Simplicidade		21
Coragem		21
Planejamento		22
Projeto.....		23
Codificação.....		23
Testes		23
5.	RESULTADOS	25
5.1.	Requisitos.....	25
5.2.	Boas Práticas.....	34
5.2.1.	Práticas trabalhadas durante o desenvolvimento.....	35
5.2.1.1.	Práticas organizacionais	35
5.2.1.2.	Práticas de equipe.....	36
5.2.1.3.	Práticas em pares.....	38
5.3.	Aplicação de Testes.....	43
5.3.1.	Testes Funcionais	43
5.3.1.1.	Teste Unitário.....	43
5.3.1.2.	Teste de Integração.....	44
5.3.2.	Teste de Aceitação.....	44
5.4.	Implementação do Protótipo: dificuldades, facilidades e requisitos futuros.....	44

6. CONSIDERAÇÕES FINAIS	47
7. REFERÊNCIAS BIBLIOGRÁFICAS	49
ANEXOS	52
APÊNDICES	55

1. INTRODUÇÃO

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 atender a esse problema, surge então a monitoria acadêmica que, de acordo com a Lei nº. 9394/96 no artigo 84 estabelece que:

Os discentes da educação superior poderão ser aproveitados em tarefas de ensino e pesquisa pelas respectivas instituições, exercendo funções de monitoria, de acordo com seu rendimento e seu plano de estudos.

Ela é de suma relevância para a formação acadêmica dos alunos, pois, quando realizada de maneira eficaz, reduz os índices de reprovação e evasão nas instituições de ensino, além de estimular o ensino-aprendizado do aluno-monitor. Tais acadêmicos, para exercerem suas funções na monitoria, necessitam fazer a revisão do conteúdo da disciplina estudada em sala, para ser possível retransmiti-los para os demais alunos que necessitam de reforçar e sanar suas dúvidas sobre os tópicos em questão, fazendo com que dediquem mais tempo aos estudos.

O monitor é um aluno, participa da cultura própria dos alunos, que tem diferenças com a dos professores. A interação daquele com a formação dos alunos da disciplina tende a favorecer a aprendizagem cooperativa, contribuindo com a formação dos alunos e do próprio monitor. (NUNES, 2005, p.53)

Mas ela não é tão simples como apresenta ser, pois necessita de um processo seletivo para escolher os alunos que possuem a disponibilidade e os pré-requisitos para se tornar monitor, e eles têm que fazer relatórios, lista de frequência e buscar métodos que os auxiliem durante a monitoria de acordo com o “ANEXO A”. Além disso, ao final do semestre, a coordenação de monitoria revisa os relatórios e calcula as horas de monitorias dadas para emissão de certificados. São muitos processos que demandam tempo e trabalho da coordenação, dos professores e também dos alunos para que ela seja executada de forma eficaz.

Neste contexto, quais as funcionalidades são necessárias para o gerenciar o ambiente da coordenação de monitoria, dos docentes, monitores e alunos de forma a facilitar o acesso e a comunicação entre eles garantindo a qualidade e satisfação dos usuários?

Tendo em vista, será desenvolvida uma ferramenta que irá auxiliar todo este processo, a fim de economizar tempo de todos os envolvidos. Um *software* que facilitará todas as etapas, desde o cadastro e seleção de monitores até a emissão de certificados ao final da monitoria. Para garantir as boas práticas de desenvolvimento e qualidade, será adotado o processo de engenharia de desenvolvimento ágil de *software*, pois este método garante uma melhor

comunicação com o cliente, um foco maior no protótipo e uma maior facilidade de aplicação de alterações. Sendo assim, de acordo com Pressman (2011, p.81):

A engenharia de software ágil combina filosofia com um conjunto de princípios de desenvolvimento. A filosofia defende a satisfação do cliente e a entrega incremental antecipada; equipes de projeto pequenas e altamente motivadas; métodos informais; artefatos de engenharia de software mínimos; e, acima de tudo, simplicidade no desenvolvimento geral.

Com isso, o aluno terá a oportunidade de recuperar o aprendizado perdido pois, de acordo com o artigo da NATÁRIO e SANTOS (2010):

A monitoria propicia mais um espaço para o aluno discutir suas dúvidas, fazer ou refazer exercícios, experimentos e assim ter sua aprendizagem mediada pelo monitor, que, por sua vez, terá espaço de ação junto ao professor, podendo receber novos textos, experimentos e realizar discussões, alicerçando, dessa forma, seus conhecimentos e construindo novas sínteses relevantes para o desempenho de suas funções e formação acadêmica.

Observando-se então o benefício mútuo do aluno com o aprendizado, do monitor recebendo certificado de horas curriculares provindos do exercício da monitoria, do professor tendo uma ajuda extra na disseminação do conhecimento e da coordenação de monitoria que terá todo esse processo mais facilitado em questão da publicação de edital, seleção de monitores e emissão de certificado. Diante do exposto, o *software* mostrará uma melhora em todo o *software* de monitoria, gerando pontos positivos para todos os envolvidos no processo.

2. OBJETIVOS

2.1. Objetivo Geral

Desenvolver um *software* de monitoria para a gestão e execução da mesma nos Bacharelados em Computação do Centro Universitário de Anápolis - GO.

2.2. Objetivos Específicos

- Identificar os requisitos, propondo uma arquitetura que garante a qualidade e as boas práticas de desenvolvimento.
- Aplicar um processo ágil para o desenvolvimento e testes do *software* em questão.
- Assegurar a qualidade do protótipo desenvolvido, colocando-o em execução.

3. METODOLOGIA

A estratégia metodológica proposta consistiu de uma pesquisa exploratória com conhecimento aprofundado da técnica de processo ágil que utilizada no desenvolvimento do *Software* de Monitoria projetado pelos autores deste, e a escolha dos referenciais teóricos em que as pesquisas foram embasadas.

Para o desenvolvimento deste trabalho, foi decidido fazer uma abordagem qualitativa, uma categoria de pesquisa flexível com foco na área de coleta e análises de dados, expectativa dos interessados e recursos que estes possam oferecer e que são planejadas para um aspecto aproximativo de determinado fato que pode ocorrer antes, durante ou após, Gil (2018).

Posteriormente, para o desenvolvimento do *software*, foi aplicado o método de processo ágil de programação extrema, que contempla um critério variante de mudanças constantes e requisitos vagos, sendo ideal para equipes pequenas e médias abrangendo cinco valores básicos que gera alguns princípios, segundo Teles (2014). Visando a criação do protótipo com uma qualidade melhor, onde sua construção é mais rápida que os demais processos e também mais econômica, garantindo a satisfação do cliente. Proporcionando um ambiente melhor conduzido em quatro valores: comunicação, simplicidade, *feedback* e coragem, conforme Teles (2014).

O estudo foi desenvolvido com a pretensão de trabalhar duas vezes por semana na construção do *software* e na metodologia a ser aplicado em seu processo.

Em sequência, foram levantados os requisitos e o ajuste da arquitetura para atendê-los e também às necessidades do *Software* de Monitoria, usando como base o Regimento de Monitoria do curso de Computação e o seu processo que descreve todo o funcionamento como regras a serem seguidas, formas de avaliação do aluno-monitor, dentre outros.

Em seguida, aplicou-se a Programação Extrema agregando um conjunto de técnicas que sejam coerentes e adotam dinâmicas de grupo. Segundo Pressman (2011) ela é conceituada com desenvolvimento abrangente em ambientes orientado a objetos, sendo definido por quatro regras: “planejamento, projeto, codificação e testes”. E no desenvolvimento do *software* foi aplicado este processo juntamente com testes de aceitação, funcionalidades e unidades.

Após a aplicação do processo ágil, foi analisado se o resultado final do *software* foi satisfatório através do teste de aceitação.

3.1. Identificação dos requisitos, e propor uma arquitetura que garante a qualidade e as boas práticas de desenvolvimento

Este documento terá como base todos os requisitos necessários para o desenvolvimento do *software* através das histórias de usuário. Para a criação deste, realizaram-se reuniões programadas com a coordenação de monitoria, o principal cliente, tendo assim uma noção de todo o processo de monitoria e obtendo acesso ao regimento da mesma para o levantamento dos requisitos a fim de garantir a satisfação do cliente, e para tudo dar certo foi necessário adotar a prática de cliente presente.

No levantamento dos requisitos foi utilizado formas ágeis como histórias de usuários para descrever os artefatos do *software*, sendo elas constatadas com valor para o usuário ou para o cliente. A Figura 1 mostra o modelo adotado pela equipe para a sua criação.

Figura 1: Modelo de histórias de usuários e critérios de aceitação.

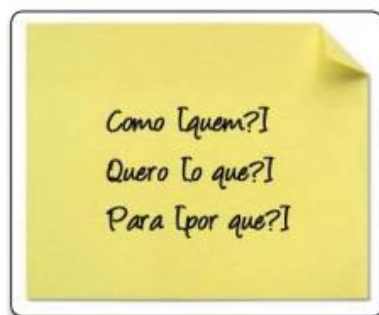


Figura 2. Modelo de cartão

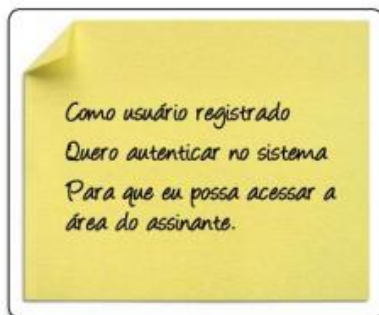


Figura 3. Exemplo de cartão

Modelo dos critérios de aceitação

- Dado [algum contexto inicial];
- Quando [algum evento ocorrer];
- Então [verificar resultados];

Exemplo de história de usuário e critério de aceitação

- **Funcionalidade:** Autenticação do assinante;
- **Como** assinante;
- **Quero** autenticar no sistema;
- **Para** que eu possa acessar o conteúdo do assinante.

- **Cenário:** Acesso a área do assinante
 - Dado que estou na tela de autenticação;
 - E possuo um usuário registrado;
 - E possuo uma assinatura válida;
 - Quando informo usuário
 - E informo senha;
 - E clico em autenticar;
 - Então visualizo a área do assinante.

Fonte: Engenharia de Software Magazine – Ed.58

Para modelagem, documentação e visualização dos requisitos os seguintes diagramas UML foram utilizados para ajudar a documentar e modelar o *software*, que para suas elaborações utiliza-se da ferramenta *Astah Community*. Segundo Larman (2007, p. 39), “(...) a UML é a notação diagramática padrão, de fato, para desenhar ou apresentar figuras (com algum texto) relacionadas a *software* – principalmente *software OO*.”.

- Diagrama de Atividades: representa as atividades que podem ser executadas pelo *software* ou pelo ator.
- Diagrama de Caso de Uso: criado para demonstrar como se relacionarão as funcionalidades do *software* e as ações de seus atores internos e externos.

E na fase estrutural, serão utilizados os:

- Diagrama de Classes: onde é mostrada as classes dentro do *software*, atributos e operações e a relação entre cada uma delas.
- Diagrama de Objetos: representa a relação entre os objetos em tempo de execução retratando o *software* em um certo momento.

Já os requisitos em relação a aplicação, foi elaborado um documento de requisitos não-funcionais, com a finalidade de informar como as funcionalidades do projeto serão entregues ao usuário final da aplicação.

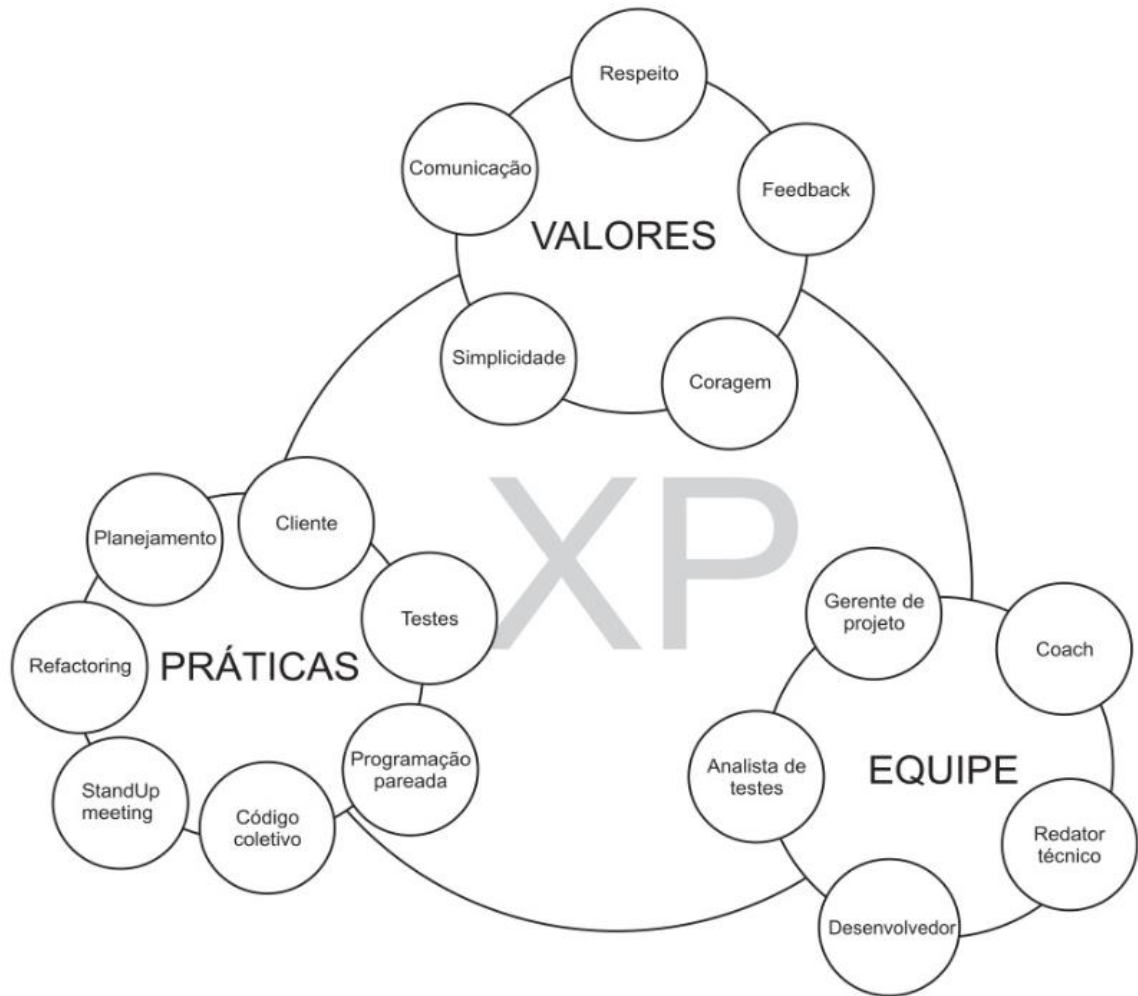
Para uma melhor qualidade dos requisitos fornecidos aos desenvolvedores do protótipo, foi esboçado com um design simples para os protótipos das telas do *software* com o intuito de evitar mudanças que possa gerar custos elevados e estouro do prazo de entrega deste.

Pretendendo ter uma melhor qualidade dos artefatos, foi utilizado o modelo de processo Prototipação para registro, evolução e validação dos requisitos para dar uma segurança maior no desenvolvimento do protótipo.

3.2. Aplicação do processo ágil para o desenvolvimento e testes do *software* em questão

Aplicar a Programação Extrema como método ágil para garantir a qualidade do protótipo e manter as boas práticas, valores e equipe durante o desenvolvimento iterativo desta etapa, conforme a Figura 2. O cliente tem um maior envolvimento na construção do *software*, sempre presente e disponível quando necessário.

Figura 2: Estrutura do Processo XP.



Fonte: SBROCCO; MACEDO, 2012.

É um processo apropriado para pequenas ou medias equipes, considerado como um método leve focado no desenvolvimento iterativo e com *releases* curtas garantindo entregas constantes de parte das funcionalidades do projeto. Esses tempos de entregas são definidos no planejamento onde o projeto foi dividido em ciclos (iteração) que estipulam quantas semanas serão necessárias para o desenvolvimento da aplicação e entrega do ciclo, gerando então um círculo até o final do projeto, conforme Sbrocco e Macedo (2012).

A codificação foi feita como um coletivo entre os desenvolvedores, onde o código fonte não possui dono e sem a necessidade de pedir para modificá-lo, tendo o objetivo de a equipe conhecer todas as partes do projeto. Porém, toda essa permissão concedida ao time, tornam eles responsáveis por qualquer alteração, segundo Teles (2014).

Outra prática adotada pela equipe foi a integração contínua, faz com que cada nova funcionalidade implementada seja integrada ao repositório no GitHub para que os códigos

sempre estejam atualizados. Tendo assim, um *feedback* sobre as alterações em menor tempo para time.

O GitHub é uma ferramenta open-source para que desenvolvedores possam hospedar seus projetos, ele tem uma grande semelhança com os serviços de nuvens por funcionar basicamente nela, tornando-o possível em armazenar os códigos-fonte e demais materiais fazendo com que o projeto seja acessado de qualquer lugar, conforme trata Pereira, Sartori e Krotz (2016).

O projeto foi desenvolvido em programação em pares, onde qualquer código seja implementado por duas pessoas na mesma máquina tendo assim melhor inspeção do que acontece. Este método de desenvolvimento é excelente para que todo o time fique a par de cada função implementada, pois há o revezamento de toda equipe durante esta etapa e em qualquer funcionalidade, de acordo com Prikladnicki, Willi e Milani (2014).

3.3. Assegurar a qualidade do protótipo desenvolvido

Sommerville (2016) diz, que o teste de *software* é o processo de executar e estabelecer se um produto atingiu todas as funcionalidades como seu funcionamento e especificações de maneira correta para o cenário o qual ele foi projetado. Na parte de testes do *software*, foram executados os testes de aceitação, unitário e integração por melhor se adequar ao projeto.

Segundo Pressman (2011), as funcionalidades de um *software* é o que satisfaz as necessidades que foram explícitas e implícitas pelo cliente, justificando um determinado contexto de uso. Foi uma maneira de testar as aplicações do *software*, verificando se a aplicação estava preparada para realizar as funções que foram programadas para fazer.

Nos testes de aceitação serão testadas determinadas funcionalidades diagnosticando sua aceitação ou não quando realizados, Pressman (2011). É considerado o mesmo teste de sistema, que tem o objetivo a garantia que o projeto esteja funcionando num todo, ele é um teste de caixa preta por ser executado com tudo conectado.

Realizado de maneira isolada, foi testado cada unidade por vez do *software* simulando possíveis dependências que a unidade testada possa apresentar. Para a realização do teste unitário, foi necessário a utilização do JUnit com o objetivo de escrever testes repetitivos que foram automatizados e com suporte em Java.

Realizado após o teste de unidade, o teste de integração é a fase em que testa a integração entre duas partes do *software*, garantindo que as classes interagem de maneira sucedida com os serviços *Web*.

Assim, os testes de aceitação, unitário e integração foram aplicados antes, durante e depois do desenvolvimento e na codificação para evitar erros, falhas e o retrabalho, fazendo com que não gerasse custos extras.

4. REFERENCIAL TEÓRICA

Esta sessão contempla os conceitos fundamentais dos elementos problemas relacionados ao escopo deste estudo: Monitoria, processo ágil, programação extrema e as suas práticas.

4.1. Monitoria

Segundo Matoso (2013), a monitoria tende a ajudar com as dificuldades encontradas na sala de aula buscando por critérios que possam abrandá-las, pois é uma forma de auxiliar aos alunos com o aprendizado e desenvolver melhor o seu raciocínio.

O exercício na monitoria revelou repercussão no desenvolvimento da relação interpessoal, quando o acadêmico-monitor mostra um melhor relacionamento com seus colegas de turma e outros grupos, a partir do desenvolvimento de trabalhos em grupo, convívio com pessoas diversificadas, tornando-se mais tolerante, respeitador das diferenças, da pluralidade[...]. (ABREU et al., 2014, p.511).

Durante a produção do *Software* de monitoria, é necessário adotar um método ou processo de desenvolvimento que auxilie os indivíduos a desenvolver algo com qualidade, e o manifesto ágil acaba se destacando devido a sua estrutura simples e funcional, possuindo vários modelos diferentes entre si, mas seguindo o mesmo conceito de agilidade.

4.2. Processo Ágil

De acordo com Sbrocco e Macedo (2012, p. 88):

O manifesto ágil foi criado considerando a velocidade demandada para novos sistemas de software, com o objetivo de abandonar métodos antigos que se mostravam ultrapassados devido ao uso de hardwares mais avançados, linguagens de programação, ambientes de desenvolvimento e necessidades organizacionais.

Segundo Prikladnicki, Willi e Milani (2014, p. 5) “Ao contrário de outras culturas de desenvolvimento, agilidade não está relacionada à obediência de protocolos preestabelecidos de produção, mas a novos padrões de comportamento e atitude”. E logo mais eles completam dizendo que “Cada Método Ágil define suas próprias práticas, mas todos, em um momento ou outro, compartilham dos valores e princípios postulados pelo Manifesto Ágil”. Dentre todos os métodos existentes, o que se encaixa no projeto devido ao foco maior no desenvolvimento foi o XP (*Extreme Programming*), que busca uma interação melhor com o cliente, desenvolvendo aos poucos com enfoque no executável.

4.2.1. Programação Extrema

Alguns autores descrevem que:

O XP é um processo de desenvolvimento que busca garantir que o cliente receba o máximo de valor de cada dia de trabalho da equipe de desenvolvimento. Ele é organizado em torno de um conjunto de valores e práticas que atuam de forma

harmônica e coesa [...]. O XP se baseia em quatro valores fundamentais: feedback, comunicação, simplicidade, coragem. (TELES, 2014, p.1)

Já Pressman (2011, p. 88) fala que a Programação Extrema define quatro regras citando-as como: “planejamento, projeto, codificação e testes”, voltadas para o ambiente orientado a objetos.

4.2.1.1. Práticas da Programação Extrema

Feedback

O *feedback* entre o cliente e a equipe de desenvolvimento gera um aprendizado simultâneo, onde o usuário consegue enxergar melhor o *software* e auxilia os desenvolvedores ao expressar suas ideias para gerar um *software* que atenda às suas necessidades.

Quando o cliente aprende com o sistema que utiliza e reavalia as suas necessidades, ele gera feedback para a equipe de desenvolvimento. Isto é, ele realimenta a equipe com alterações nas necessidades que ainda serão implementadas e, eventualmente, naquelas que já fazem parte do software. O feedback é o mecanismo fundamental que permite que o cliente conduza o desenvolvimento diariamente para aquilo que irá gerar mais valor. (TELES, 2014, p.1)

Mas tão importante quanto este valor é a interação em equipe, quando se mantém a comunicação de forma que todos trabalhem em conjunto.

Comunicação

“A comunicação entre o cliente e a equipe permite que todos os detalhes do projeto sejam tratados com a atenção e a agilidade que merecem” (TELES, 2014, p.1), evitando contratempos advindos da falta de comunicação, melhorando no desenvolvimento que é embasado na simplicidade, ou seja, desenvolver somente o necessário.

Simplicidade

“É necessário que a equipe compreenda e utilize o valor da simplicidade, que nos ensina a implementar apenas aquilo que é suficiente para atender a cada necessidade do cliente” (TELES, 2014, p.1). Ao abordar esta regra, pode-se evitar um *software* sobrecarregado de funções que possam gerar lentidão e futuros transtornos, encarando com coragem cada etapa de processo.

Coragem

“A equipe precisa ser corajosa a acreditar que, utilizando as práticas e os valores do XP, será capaz de fazer o *software* evoluir com segurança e agilidade” (TELES, 2014,

p.1). Acreditar também que ao seguir o planejamento proposto, será capaz de desenvolver algo com qualidade.

Planejamento

Os requisitos são especificações do que o *software* deve fazer, os seus serviços a serem oferecidos e as restrições de seu funcionamento, tendo como finalidade de representar as necessidades do cliente no *software* como encontrar informações, conforme Sommerville (2011).

“Uma atividade de levantamento de requisitos que capacita os membros técnicos da equipe XP a entender o ambiente de negócios do *software* e possibilita que se consiga ter uma percepção ampla sobre os resultados solicitados, fatores principais e funcionalidade” (PRESSMAN, 2011, p. 88). Uma equipe bem preparada é fundamental no desenvolvimento de *softwares*, pois conseguem enxergar o *software* como um só, conseguindo prever erros e melhorando drasticamente a qualidade e economia de tempo durante o desenvolvimento, garantindo o sucesso do projeto.

Segundo Sommerville (2011, p.57), “Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços oferecem e as restrições a seu funcionamento”. Elaborase um documento de requisitos em que irá delimitar o escopo do *software* de monitoria provendo suas funcionalidades, descrevendo quais os atributos de qualidades serão suportados.

No desenvolvimento de um site, é preciso saber que as aplicações Web são *software* necessitantes da internet como ambiente para sua execução. E para entender esse tipo de *software*, é necessário compreender seus requisitos e esquematizar as principais atividades, conforme Miletto e Bertagnolli (2014).

Segundo Cunha, Figueira e Viana (2017), uma arquitetura de *software* pode ser considerada um plano inicial e abrangente de seus componentes. A arquitetura proposta foi desenvolvida com base em camadas, tornando possível dividir o *software* em módulos que se comunicam entre as camadas.

Pressman (2016) afirma que, um processo não é uma maneira de como o *software* deve ser desenvolvido, no entanto, pode ser considerado como uma forma adaptável que permite a equipe realizar o trabalho de eleger o grupo de ações e tarefas que forem apropriadas para o *software*.

Projeto

“O projeto XP segue rigorosamente o princípio KIS (*keep it simple*, ou seja, preserve a simplicidade). É preferível sempre um projeto simples do que uma representação mais complexa” (PRESSMAN, 2011, p. 89). Este ponto é algo que tem em comum entre Teles e Pressman, pois a Programação Extrema tem como objetivo desenvolver o mais simples possível e futuramente implementar modificações, alterações e outras funções quando solicitadas.

Sommerville (2011) descreve que na Programação Extrema, os requisitos dos *softwares* são expressos em cenários que são descritos COMO, QUERO e PARA, estas histórias de usuários são implementadas diretamente como uma série de tarefas. Na parte do desenvolvimento, os trabalhadores são instruídos a trabalharem em pares e sempre fazem os testes antes de começarem a codificar. Assim, sempre que um novo código for acoplado ao *software*, os testes devem ser executados com sucesso, tendo assim um pequeno intervalo entre as *releases*.

Codificação

Pressman (2016) define que manter a interface, aplicação e navegação de maneiras distintas por camadas (MVC), é uma forma de facilitar a implementação que acaba gerando um aumento de reutilização que leva a boas práticas

Depois de desenvolvidas as histórias e o trabalho preliminar de elaboração do projeto ter sido feito, a equipe não passa para a codificação, mas sim, desenvolve uma série de testes de unidades que exercitarão cada uma das histórias a ser inclusas na versão corrente [...]. Uma vez criado o teste de unidades, o desenvolvedor poderá melhor focar-se no que deve ser implementado para ser aprovado no teste. (PRESSMAN, 2011, p. 90).

Testes

Segundo Sommerville (2011), nos testes funcionais o testador só se preocupa com a funcionalidade, deixando de lado a forma que foi implementada. Dentre estes testes pode-se citar os testes de unidade e de integração.

“Já foi observado que a criação de testes de unidade, antes de começar a codificação, é um elemento-chave na abordagem XP. Os testes de unidade criados devem ser implementados usando-se uma metodologia que os capacite a ser automatizados” (PRESSMAN, 2011, p. 90). Ele testa pequenas partes de projeto individualmente, a fim de descobrir erros e falhas dentro dos limites do módulo (PRESSMAN, 2016, P.473). Já o teste de integração, que tem como objetivo testar a interação entre as partes do projeto. “O objetivo é construir uma estrutura de programa determinada pelo projeto a partir de componentes testados em unidade”

(PRESSMAN, 2016, p. 476). E além destes, será utilizado o teste de aceitação que verificará se o *software* desenvolvido atende as necessidades do cliente, contribuindo para a avaliação do protótipo.

O teste de aceitação tem por finalidade revelar erros e omissões na definição dos requisitos do *software*. Podem revelar também problemas nos requisitos do *software* onde os recursos podem não atender as necessidades do usuário ou seu desempenho seja inaceitável. Ele é feito no final do processo de testes, antes que o *software* seja aceito para uso operacional, referindo-se Sommerville (2011).

5. RESULTADOS

5.1. Requisitos

Trata-se de um *software* que permita a iteração de aluno, aluno-monitor, professor e administrador. Para que isto aconteça, foi necessário adotar uma metodologia ágil para o desenvolvimento do *software*. Existem diversas funcionalidades dentro deste, fazendo com que haja iteração entre esses usuários.

No desenvolvimento, foi necessário obter alguns requisitos do *software* para que fosse implementado. Até então, desenvolveu-se as Histórias de Usuários, que mostram os objetivos que devem ser seguidos e os seus cenários, que é o caminho para atingir um determinado objetivo. Dentre as histórias produzidas, selecionou-se a “Emitir Ficha de Avaliação” para ser aqui inserida. As demais encontram-se no Apêndice A, onde as já desenvolvidas se encontram com as telas e funcionalidades.

[EMITIR FICHA DE AVALIAÇÃO] Emitir Ficha de Avaliação

COMO professor do sistema

QUERO emitir ficha de avaliação de monitoria

PARA avaliar o monitor e os alunos que participam das monitorias

CENARIO 1: Emitir ficha

DADO que estou logado no sistema

QUANDO clico em Emitir ficha de Avaliação

E abre a ficha de avaliação

E insiro os dados necessários nos campos

E clico em Emitir

ENTÃO é emitido as fichas de avaliações

CENARIO 2: Erro ao emitir ficha

DADO que estou logado no sistema

QUANDO clico em Emitir ficha de Avaliação

E é aberta a ficha de avaliação

E insiro os dados necessários nos campos

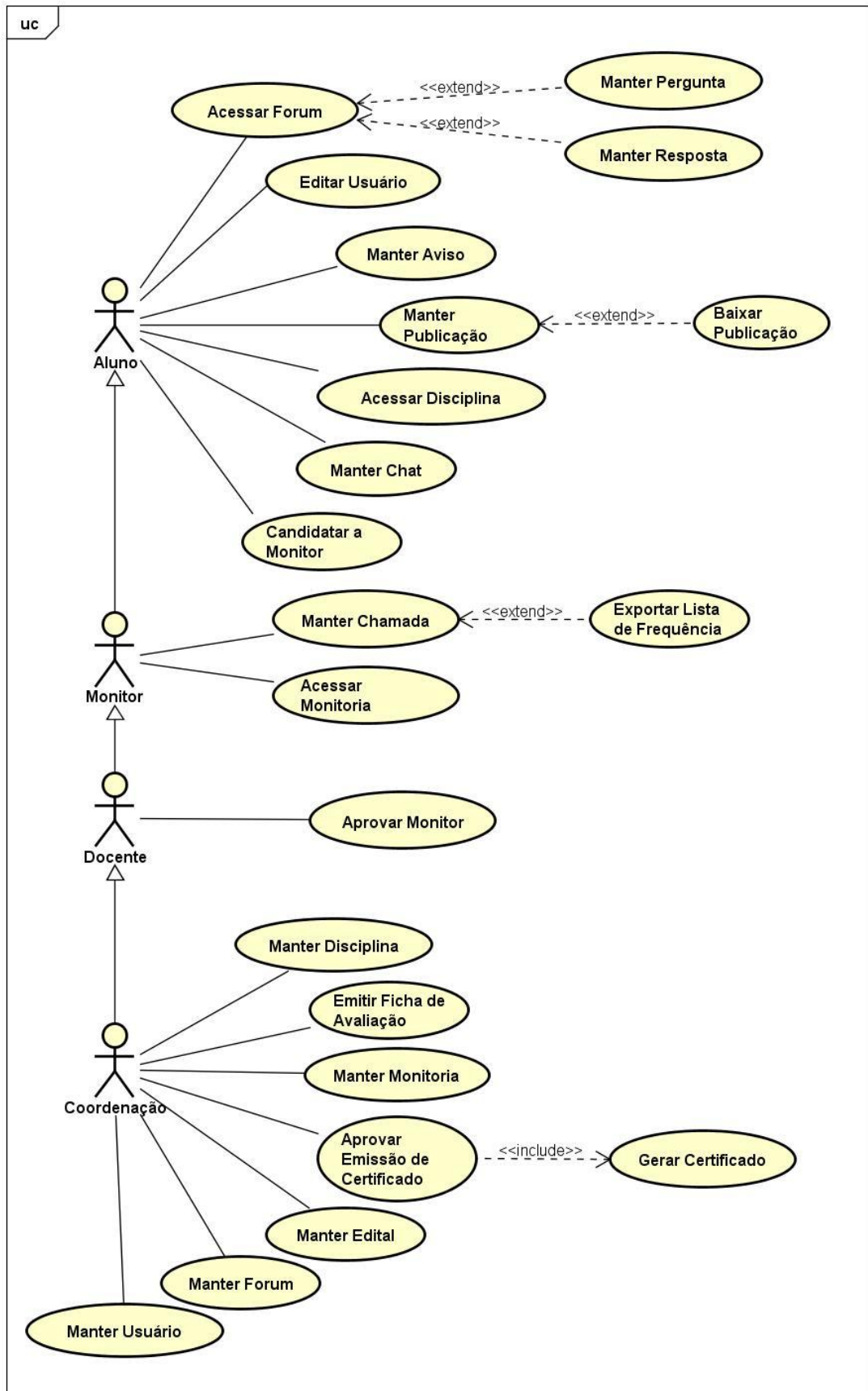
E deixo algum campo obrigatório em branco

E clico em Emitir

ENTÃO exibe uma mensagem informando o erro

Através das Histórias de Usuários foi possível desenvolver o Diagrama de Caso de Uso para mostrar as funcionalidades que cada ator possa executar como é mostrado na Figura 3. Em que, alguns atores têm tarefas por heranças que são herdadas de outros e tarefas que são atribuídas somente a ele, possibilitando ter uma visualização da implementação antes de ser realizada proporcionando uma melhor visão do protótipo.

Figura 3: Diagrama de Caso de Uso



Fonte: (Os Autores, 2018)

Logo após, os requisitos não-funcionais foram identificados estando relacionados a aplicação em alguns termos, que diz respeito a como estas aplicações devem ser entregues ao usuário final do *software*, conforme a Tabela 1. Os demais requisitos não-funcionais estão disponíveis no Apêndice B.

Tabela 1: Requisito Não-Funcional.

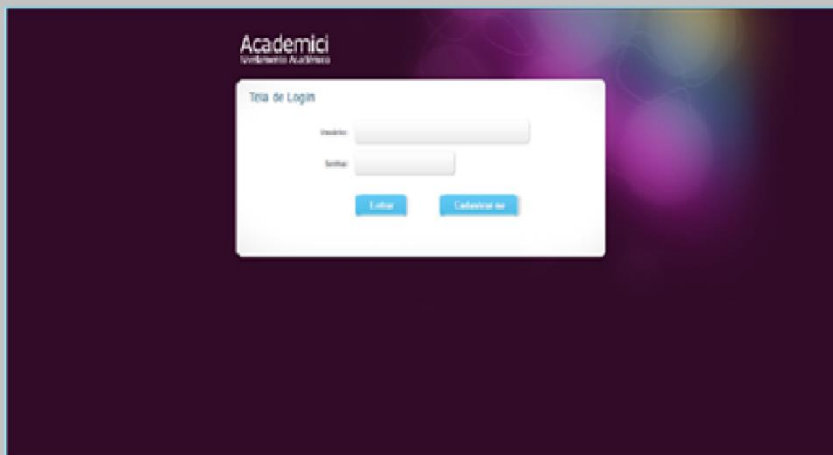
Identificador	RNF002	Categoria	Segurança
Nome	Autenticação de usuário		
Data de Criação	17/09/2018	Autor	Wanderson Inácio
Data da última alteração	N/A	Autor	N/A
Versão	1.0	Prioridade	Alta
Descrição	<p>Para a autenticação de cada usuário para acessar o nível de aplicação, é necessário que tenha um cadastro válido na base de dados do sistema.</p> <p>Deverá haver uma política de segurança que assegure a autenticidade de cada um dos usuários.</p>		

Fonte: (Os Autores, 2018)

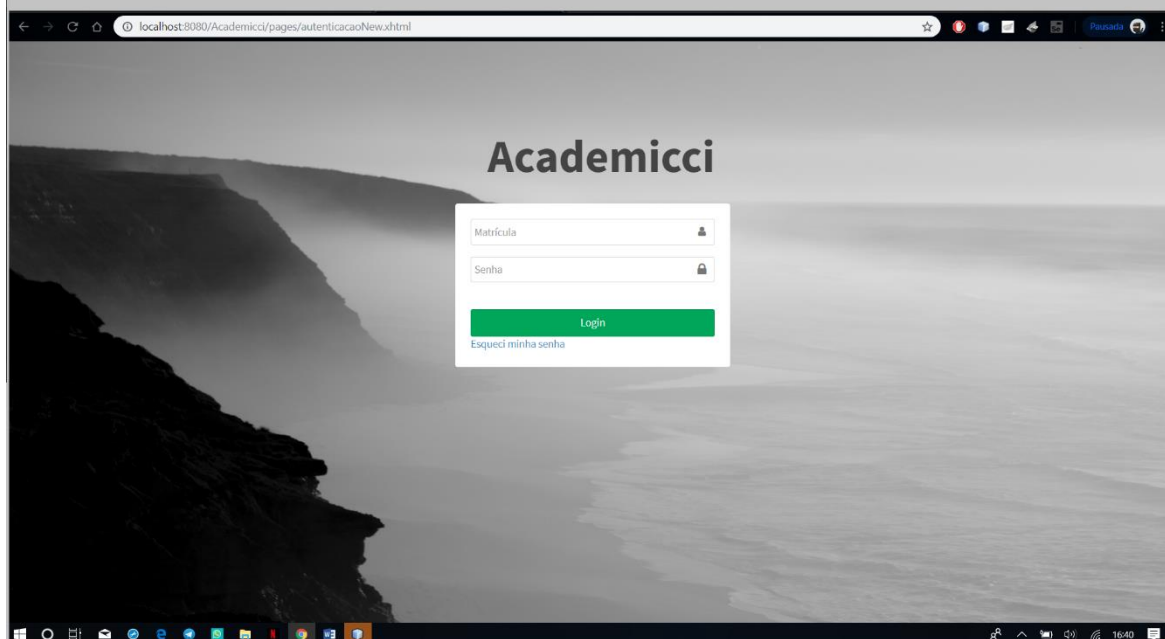
Os protótipos das telas foram esboçados em um fundo com cor quente, e suas telas eram trabalhadas em cores frias como mostra a Figura 4. Esse protótipo foi utilizado por um tempo pela equipe de desenvolvimento, até que encontrou dificuldades nesse *template* e resolveu adotar um novo *layout* que se adequasse melhor ao projeto como mostra na imagem.

Figura 4: Protótipos utilizado no projeto

1º Protótipo da Tela de Login



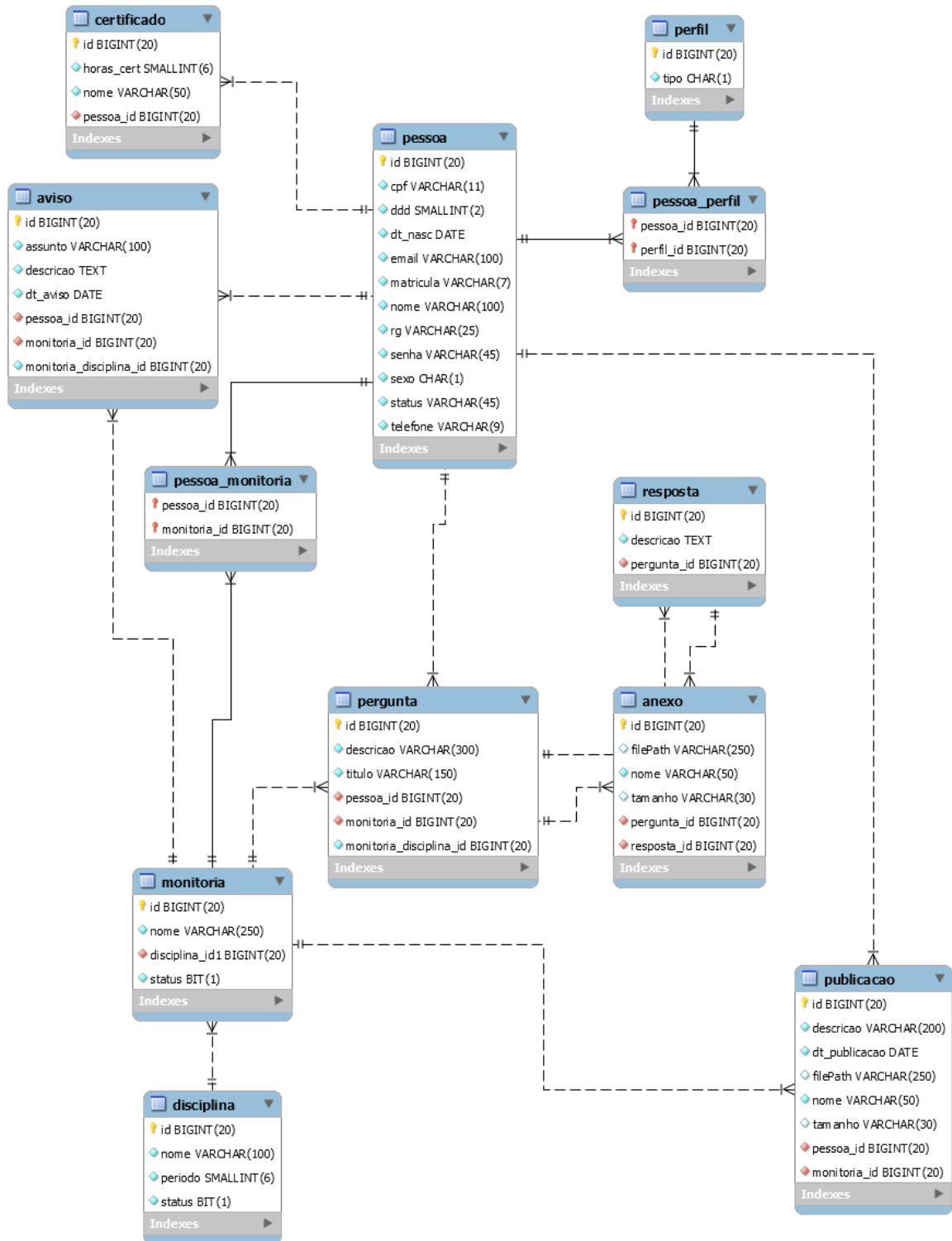
Modelo final da Tela de Login



Fonte: (Os Autores, 2017.)

Em seguida foi desenvolvido o Modelo Lógico do MER (Modelo Entidade-Relacionamento) que especifica os relacionamentos entre as tabelas do banco de dados, de modo que seja possível ter uma ideia das interações entre as classes durante o desenvolvimento, exibido na Figura 5.

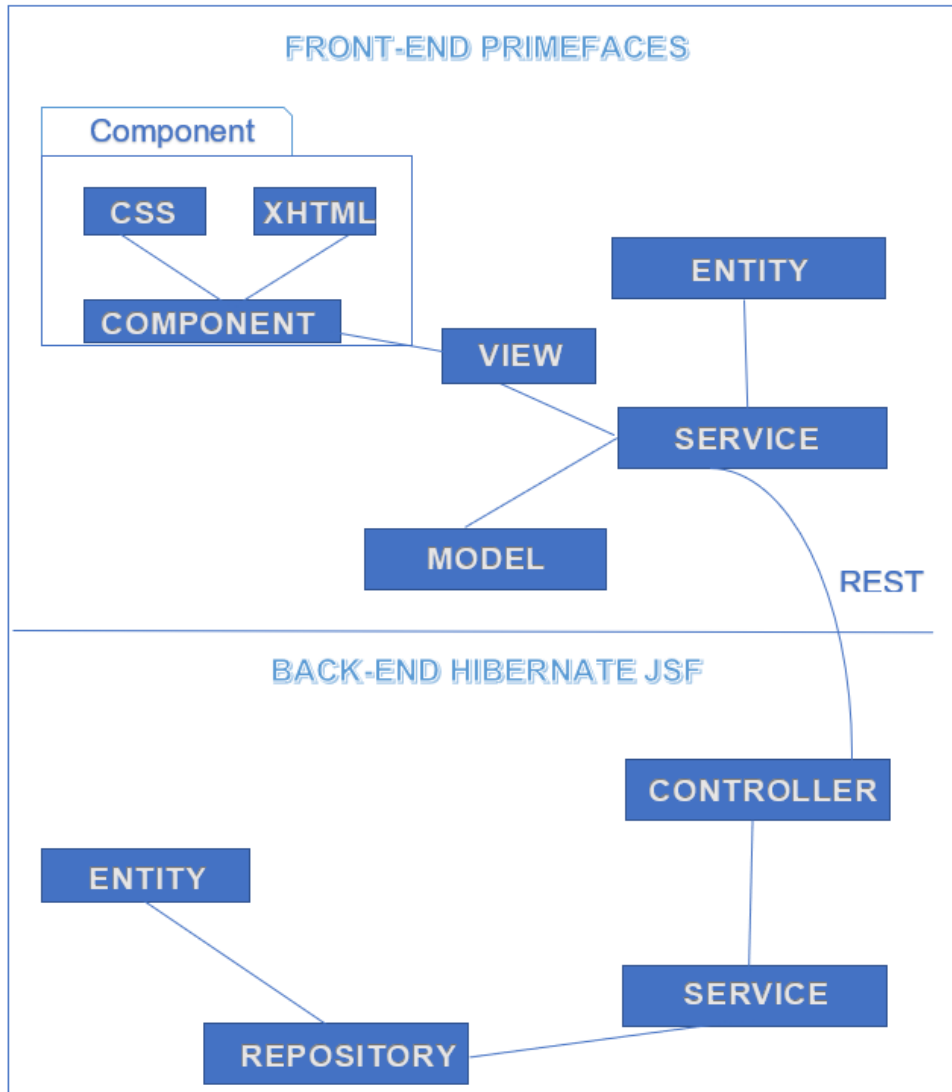
Figura 5: Modelo lógico do MER



Fonte: (Os Autores, 2018.)

A Figura 6, mostra um diagrama exibindo a visão dos componentes que formara o *software*, representando o *back-end* a parte de processamento final dos dados e o *front-end* que está por trás das interfaces de navegação.

Figura 6: Visão dos componentes



Fonte: (Os Autores, 2018)

Durante o processo de análise, a equipe decidiu desenvolver o *software* utilizando o *NetBeans* que é uma *IDE* de ambiente integrado e com código aberto na linguagem *Java Script*. Permitindo aliar essa estrutura com o *HTML*, *CSS* e *PrimeFaces* para estilizar os elementos das páginas dando um visual mais completo. Um exemplo disso é exibido na Figura 7.

Figura 7: Código fonte da tela de *login*

```

<h:body styleClass="hold-transition login-page">
  <p:growl autoUpdate="true" sticky="true"/>
  <div id="loader" class="load-bar" style="display: none">
    <div class="bar"></div>
    <div class="bar"></div>
    <div class="bar"></div>
  </div>

  <p:dialog id="adminStatusDialog" modal="true" widgetVar="statusDialog" draggable="false" closable="false"
    resizable="false" responsive="true" showHeader="false" appendTo="@{body}">
<!--   <p:graphicImage library="images" name="#{adminConfig.loadingImage}"/>-->
  </p:dialog>

  <div class="login-box">
    <div class="login-logo">
      <p:link href="index.xhtml"><b>Academicci</b></p:link>
    </div>
    <!-- /.login-logo -->
    <div class="box login-box-body">
      <h:form>
        <p:messages closable="true" />
        <div class="form-group has-feedback">
          <p:inputText value="#{autenticacaoBean.usuario.matricula}" styleClass="form-control"
placeholder="Matrícula"
            required="true" autocomplete="off" maxLength="7"
            requiredMessage="Matrícula é obrigatório."/>
          <i class="fa fa-user form-control-feedback"></i>
        </div>
        <div class="form-group has-feedback">
          <p:inputText value="#{autenticacaoBean.usuario.senha}" type="password" styleClass="form-control"
placeholder="Senha" required="true" autocomplete="off"
            requiredMessage="Senha é obrigatório."/>
          <i class="fa fa-lock form-control-feedback" style="font-size: 18px" ></i>
        </div>
        <div class="row">
          <p:spacer height="10"/>
          <div class="col-xs-12">
            <p:commandButton styleClass="btn btn-success btn-block" onclick="showBar()"
              actionListener="#{autenticacaoBean.autenticar}" oncomplete="if(args.validationFailed) {
hideBar()}"
              value="Login" process="@form" update="@form"/>
          </div>
        </div>
        </h:form>
        <a href="#">Esqueci minha senha</a><br/>
      </div>
    <!-- /.login-box-body -->
  </div>
</h:body>
</html>

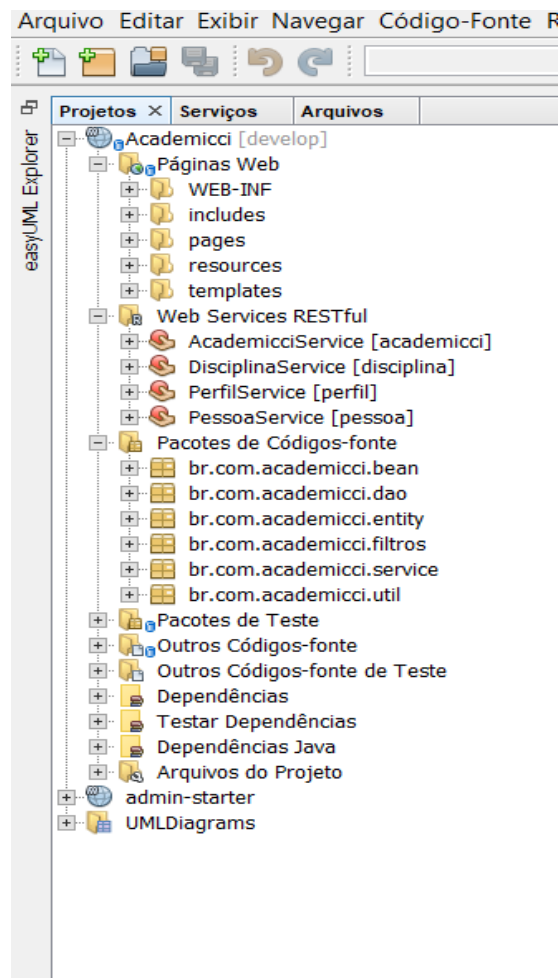
```

Fonte: (Os Autores, 2018)

O *software* foi desenvolvido utilizando na arquitetura a estrutura MVC (*Model-View-Controller*), um padrão de projeto que trabalha com aplicações de pacotes de desenvolvimento rápido, de fácil manutenção e modular. A elaboração das tarefas divididas entre as camadas

Model, View e Controller torna a aplicação mais leve, pois cada uma delas executa apenas o que lhe é definido, a Figura 8 mostra a estrutura usada no desenvolvimento do *software*.

Figura 8: Estruturação do código do *software*



Fonte: (Os Autores, 2018)

No desenvolvimento, é utilizado em sua construção de interfaces o *JavaServer Faces* (JSF) e o *PrimeFaces* que é uma biblioteca de componentes de interface, como *framework* utilizou o *Hibernate* e o *MySQL Server* como servidor de banco de dados.

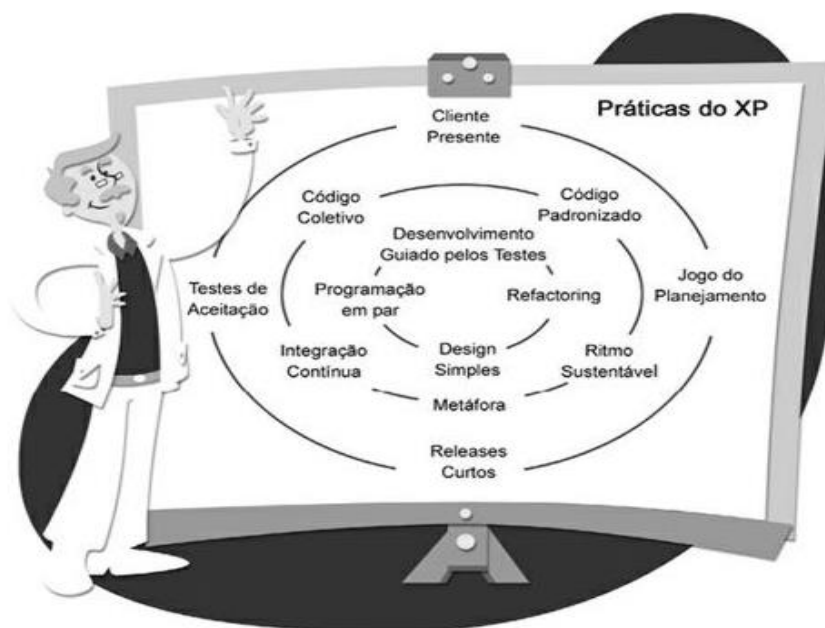
Com base em alguns documentos trabalhados na FTT, desenvolveu-se o Documento de Arquitetura que é mostrado no Apêndice C. Este documento forneceu uma visão geral da arquitetura abrangente do *software*, fornecendo diversas visões possibilitando descrever distintos tópicos do *software*.

5.2. Boas Práticas

As práticas são medidas adotadas pela equipe XP, que priorizou tudo aquilo que era feito diariamente durante o desenvolvimento do projeto, gerando melhorias em seu processo e no trabalho da equipe. Um exemplo disso era a reunião diária, realizada em pé e todos os membros da equipe participam falando o que fez no último encontro, quais os impedimentos encontrados e as próximas tarefas a fazer. Isso foi um método de boa prática, pois, através destas reuniões era possível saber como estava o andamento do projeto e as dificuldades que a equipe havia enfrentado.

No XP há três tipos de práticas que é trabalhado, cada um desses engloba tarefas que a equipe define quais ela utilizara para cada uma das práticas. Elas são divididas em práticas organizacionais o círculo maior, práticas de equipe o círculo médio e as práticas de pares o círculo pequeno com mostra a Figura 9.

Figura 9: Práticas do XP.



Fonte: (TELES, 2014).

Para o desenvolvimento do protótipo, foi adotado como:

- Práticas organizacionais: padrões de Cliente Presente, *Releases Curtos* e Testes de Aceitação;
- Práticas de equipe: padrões de Integração Contínua e Código coletivo;
- Práticas de pares: padrões de *Design Simples* e Programação em par.

Essas são as práticas trabalhadas durante o projeto pela equipe, colocando em ação cada uma destas mencionadas.

5.2.1. Práticas trabalhadas durante o desenvolvimento

No desenvolvimento do projeto, logo acima está descrito as práticas adotadas do XP desde a parte de documentação até a sua implementação. Em seguida são referidas características de cada uma destas empregadas no projeto.

Essas práticas levam à um desenvolvimento mais rápido e interativo com o cliente, recebendo *feedbacks* constantemente do que está sendo produzido pela equipe.

5.2.1.1. Práticas organizacionais

Uma das práticas adotadas pela equipe é a de cliente presente, que participou durante todo o processo desde o início ao fim, para que orientasse o desenvolvimento através de seus *feedbacks*.

Na construção dos documentos realizou-se várias conversas com o cliente no qual ele verificava se está de acordo ou não, caso esteja tudo em ordem é seguido para outro artefato, mas se o cliente não aprovar será realizado alterações e levado para ele validar novamente. Essa prática foi adotada durante toda a sua execução.

Durante o período de desenvolvimento, trabalhou-se com *releases* curtas em cada uma das *Sprints*, que determinava quantos dias ou semanas duraria o tempo de execução de uma *Sprint*. Assim, a entrega dos resultados eram mais rápidas e valorizavam o protótipo em desenvolvimento, tendo mais *feedbacks* e gerando uma maior confiança no que era desenvolvido.

Durante o desenvolvimento do *software*, a equipe trabalhou no decorrer do projeto com 7 *Sprints*, onde cada uma delas foram criadas para uma parte do projeto como mostra os objetivos na Tabela 2.

Tabela 2: *Sprints* Trabalhadas

<i>Sprints</i>	Objetivos
1 ^a	Realizar o levantamento dos artefatos e documentações do <i>software</i> .
2 ^a	Realizar o treinamento da equipe e fazer refinamento da documentação do <i>software</i> .
3 ^a	Fazer implementação back-end e refinamento da documentação do <i>software</i> .
4 ^a	Realizar implementação do banco e base das telas do <i>software</i> .
5 ^a	Fazer reestruturação do ambiente, montagem da arquitetura e refinamento dos documentos do <i>software</i> .
6 ^a	Fazer implementação da metade dos requisitos Front-end e dos testes do <i>software</i> .
7 ^a	Realizar os testes e finalização das funcionalidades do <i>software</i> .

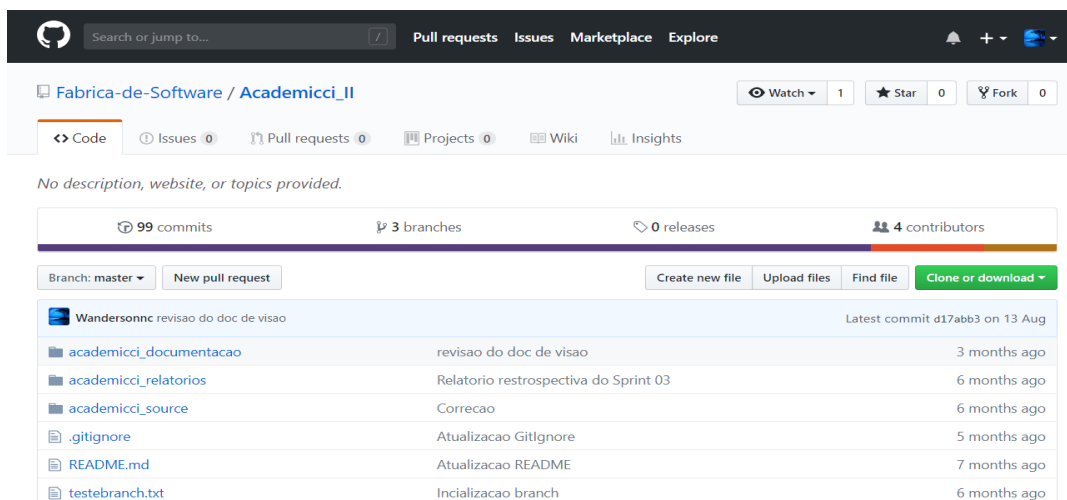
Fonte: (Os Autores, 2018).

O teste de aceitação foi aplicado ao final do processo, com a finalidade de verificar se o *software* está pronto e se pode ser usado pelos usuários para execução das funções e tarefas para o qual foram desenvolvidos.

5.2.1.2. Práticas de equipe

Uma das práticas a qual foi adotada pela equipe é a de integração contínua, que é utilizado desde o início do projeto a ferramenta Github como mostra a Figura 10.

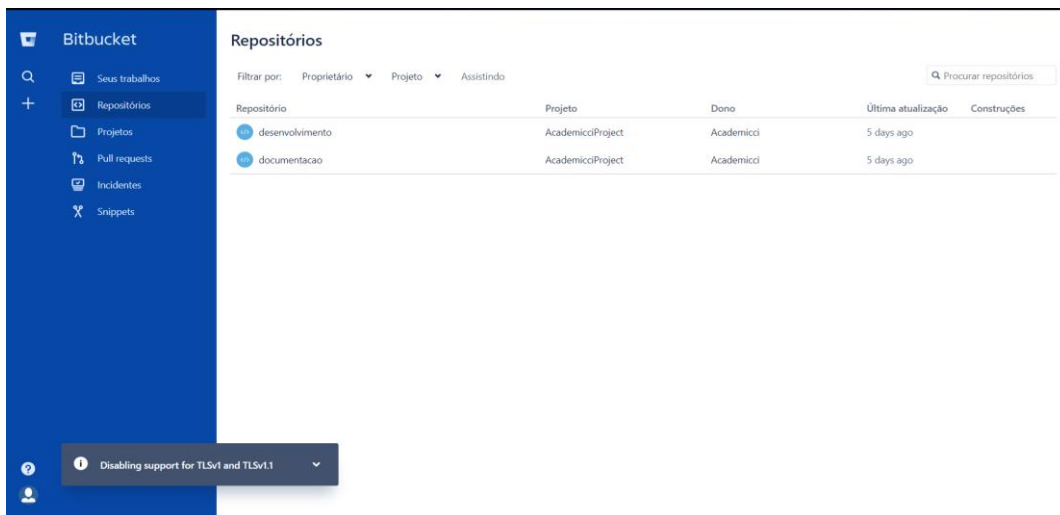
Figura 10: Ferramenta Github.



Fonte: (Os Autores, 2018)

Em seguida foi decidido usar uma extensão do Git que atendesse melhor as necessidades do projeto, então foi escolhido utilizar o Git Flow operando um serviço de hospedagem como o Bitbucket mostrado na Figura 11. Estes foram utilizadas para disponibilizar tudo o que fosse desenvolvido do projeto para outros integrantes e para gerenciar o que está acontecendo.

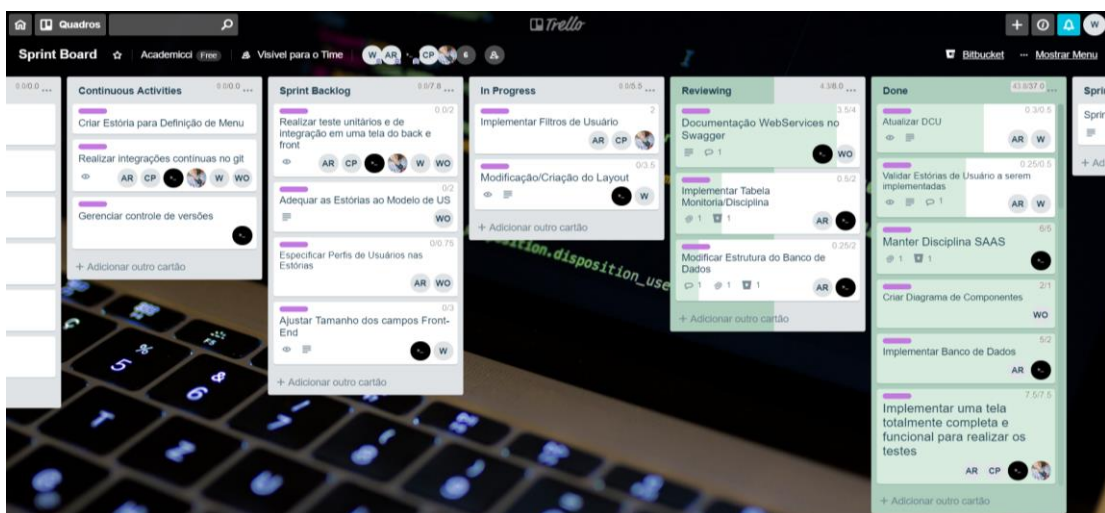
Figura 11: Ferramenta Bitbucket.



Fonte: (Os Autores, 2018).

Para o gerenciamento das atividades do projeto e das *Sprints*, foi utilizado como ferramenta o Trello que permite criar quadros virtuais para gerenciar as tarefas e atribui-las a quem fosse realizar, como mostra a Figura 12.

Figura 12: Trello – Gerenciamento de tarefas



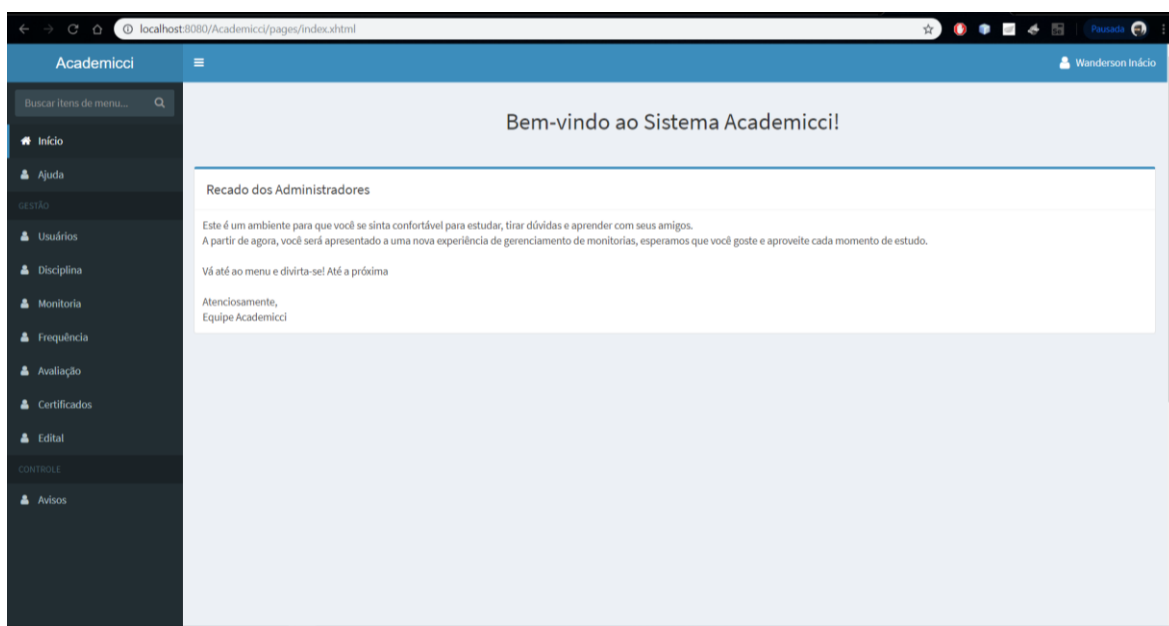
Fonte: (Os Autores, 2018).

No desenvolvimento foi usado a coletividade do código, permitindo que qualquer um da equipe pudesse realizar alterações fazendo com que todos tenham conhecimento dos componentes do *software*, porém, cada um se tornava responsável pela alteração feita.

5.2.1.3. Práticas em pares

Para o design das telas, opta-se pela prática de design simples onde o menos é mais, sem muita poluição e cores mais sóbrias para agradar aos olhos de quem vê. Cores muito chamativas e vários ícones podem prejudicar além de dificultar o acesso, tornando o *software* desagradável. A Figura 13 mostra o *design* da tela inicial, onde a barra lateral preta é retrátil, podendo ela mostrar só os ícones ou os ícones com a descrição, e há também a tela no qual aparecera as funcionalidades quando clicadas.

Figura 13: Tela inicial do *software* Academicci



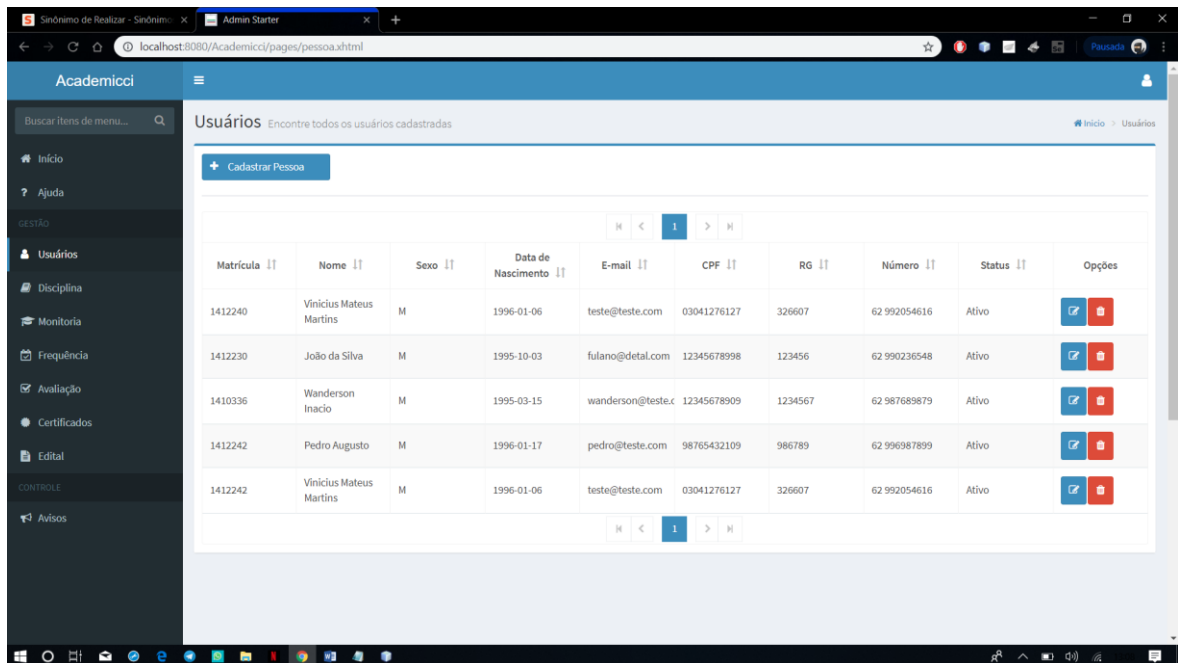
Fonte: (Os Autores, 2018).

Na implementação a equipe se dividiu em duas partes sendo *Back-end* para trabalhar por trás do projeto e *Front-end* trabalhando com a interface do *software*, e dentro desses é exercido a prática de programação em pares, cujo objetivo é todos dentro da equipe saber o que está acontecendo no projeto e programar. Essa técnica faz com que os que têm mais conhecimento dentro da equipe ajuda a sua dupla a programar e aprimorar-se mais.

Na Figura 14, é mostrado o *layout* da página de usuários onde existem todas as atividades como a de cadastro de usuário, edição e exclusão. As demais páginas desenvolvidas como monitorias, disciplinas e avisos seguem o mesmo padrão, com as diferenças apenas de

cada uma das funcionalidades. No Apêndice A é possível visualizar todas as telas já desenvolvidas.

Figura 14: Tela de Usuários do *software* Academicci



Fonte: (Os Autores, 2018).

É possível visualizar na Figura 15 o código fonte da tela de usuário, onde ela foi desenvolvida por programação em par, utilizando uma das práticas do XP. O desenvolvimento desses protótipos no início foram complicados por não ter conhecimento de *PrimeFaces*, pois era uma ferramenta nova para a equipe que nem havia ouvido falar. Então o *front-end* passou a estudar a ferramenta obtendo conhecimento para então desenvolver os protótipos.

Figura 15: Código fonte da Tela de Usuários

```

<ui:define name="metadata">
  <ui:param name="title" value="Disciplinas"/> <!-- Automatic create breadCrumb and page title when param
'title' is provided. -->

  <style type="text/css">
    .ui-datatable .ui-datatable-header {
      text-align: right !important;
    }
  </style>
</ui:define>

<ui:define name="description">
  Encontre todas as disciplinas existentes
</ui:define>

<ui:define name="body">
<!-- <p:growl id="mensagem" />-->
  <h:form id="formListagem">
    <div class="box box-primary">
      <div class="box-header with-border">
        <div id="main-buttons" class="hidden-sm hidden-xs">

          <p:commandButton icon="fa fa-plus" value="Cadastrar Disciplina" styleClass="btn-primary"
            oncomplete="PF('dialogo').show();"
            update=":formCadastro:pnCadastro"
            actionListener="#{disciplinaBean.novo}">
            <f:actionListener type="org.omnifaces.eventlistener.ResetInputAjaxActionListener" />
          </p:commandButton>

          <p:spacer width="5"/>
        </div>
        <p:separator/>

        <p:dataTable emptyMessage="Nenhum registro encontrado."
          paginator="true" value="#{disciplinaBean.disciplinas}"
          var="disciplina" rows="10" id="tabela">

          <p:column headerText="Nome" sortBy="#{disciplina.nome}">
            <h:outputText value="#{disciplina.nome}"/>
          </p:column>

          <p:column headerText="Período" sortBy="#{disciplina.periodo}">
            <h:outputText value="#{disciplina.periodo}" />
          </p:column>

          <p:column headerText="Status" sortBy="#{disciplina.statusFormatado}">
            <h:outputText value="#{disciplina.statusFormatado}" />
          </p:column>

          <p:column headerText="Opções">
            <p:commandButton icon="fa fa-edit" styleClass="btn-primary"
              actionListener="#{disciplinaBean.editar}"
              update=":formCadastro:pnCadastro"
              oncomplete="PF('dialogo').show();">
              <f:attribute name="disciplinaSelecionada" value="#{disciplina}" />
              <f:actionListener type="org.omnifaces.eventlistener.ResetInputAjaxActionListener" />
            </p:commandButton>

            <p:commandButton icon="fa fa-trash" styleClass="btn-danger"
              actionListener="#{disciplinaBean.excluir}"
              update=":formListagem:tabela">

```



```

        <p:confirm header="Confirmação" message="Deseja excluir a Disciplina?"
            icon="fa fa-exclamation-circle" />
        <:attribute name="disciplinaSelecionada" value="#{disciplina}" />
    </p:commandButton>
</p:column>
</p:dataTable>
</div>
</div>

<p:confirmDialog global="true" showEffect="fade" hideEffect="fade" styleClass="box-danger box-solid">
    <p:commandButton value="Sim" type="button" styleClass="btn-material btn-primary ui-confirmdialog-yes"
        icon="fa fa-check"/>
    <p:commandButton value="Não" type="button" styleClass="btn-material btn-danger ui-confirmdialog-no"
        icon="fa fa-close"/>
</p:confirmDialog>
</h:form>

<p:dialog header="Cadastrar Disciplina" widgetVar="dialogo"
    draggable="false" resizable="false" modal="true" closable="false" appendTo="@{body}">
<h:form id="formCadastro">
    <h:panelGrid columns="2" id="pnCadastro">
        <p:outputLabel for="nome" value="Nome:" />
        <p:inputText id="nome" maxLength="100" size="50" value="#{disciplinaBean.disciplina.nome}"
            required="true" requiredMessage="O campo 'Nome' é obrigatório!"/>

        <p:outputLabel for="periodo" value="Período:" />
        <p:selectOneMenu id="periodo" value="#{disciplinaBean.disciplina.periodo}"
            required="true" requiredMessage="O campo 'Período' é obrigatório!">
            <f:selectItem noSelectionOption="true" itemLabel="Selecione o período" />
            <f:selectItem itemValue="1" itemLabel="1º Período" />
            <f:selectItem itemValue="2" itemLabel="2º Período" />
            <f:selectItem itemValue="3" itemLabel="3º Período" />
            <f:selectItem itemValue="4" itemLabel="4º Período" />
            <f:selectItem itemValue="5" itemLabel="5º Período" />
            <f:selectItem itemValue="6" itemLabel="6º Período" />
            <f:selectItem itemValue="7" itemLabel="7º Período" />
            <f:selectItem itemValue="8" itemLabel="8º Período" />
            <f:selectItem itemValue="9" itemLabel="9º Período" />
            <f:selectItem itemValue="10" itemLabel="10º Período" />
        </p:selectOneMenu>

        <p:outputLabel for="ativo" value="Ativo: " />
        <p:selectBooleanCheckbox id="ativo"
            value="#{disciplinaBean.disciplina.status}" required="true"
            requiredMessage="O campo 'Status' é obrigatório!" />
    </h:panelGrid>

    <h:panelGrid columns="2">
        <p:commandButton value="Salvar" actionListener="#{disciplinaBean.salvar}" styleClass="btn-primary"
            update=":formCadastro:pnCadastro :formListagem:tabela"/>
        <p:commandButton value="Fechar" oncomplete="PF('dialogo').hide();" styleClass="btn-danger"/>
    </h:panelGrid>
</h:form>
</p:dialog>
</ui:define>

```

Fonte: (Os Autores, 2018).

5.3. Aplicação de Testes

5.3.1. Testes Funcionais

Os testes funcionais propostos estão sendo desenvolvidos utilizando o framework JUnit que auxiliam no desenvolvimento dos testes unitários e de integração, tendo um total de 14% de código testado até o momento como consta na Figura 16.

Figura 16: Porcentagem de código testado.

Academicci

Element	Missed Instructions	Cov.
br.com.academicci.bean		0%
br.com.academicci.entity		32%
br.com.academicci.dao		24%
br.com.academicci.service		0%
br.com.academicci.util		29%
Total	1.830 of 2.116	14%

Fonte: (Os Autores, 2018).

5.3.1.1. Teste Unitário

O Teste unitário visa testar as funcionalidades de forma independente para que caso haja erros, eles possam ser encontrados ao examinar partes menores do *software*. É basicamente isolar as funcionalidades simulando valores de entrada e retorno e analisar se os dados e comportamentos obtidos foram os esperados, conforme Figura 17.

Figura 17: Exemplo de Teste Unitário

```
@Test
public void salvarPessoaUnitario() throws ParseException {
    PessoaDAO dao = new PessoaDAO();
    Session session = Mockito.mock(Session.class);
    Transaction transaction = Mockito.mock(Transaction.class);

    Mockito.when(session.beginTransaction()).thenReturn(transaction);

    Mockito.when(session.merge(Matchers.any(Pessoa.class))).thenReturn(this.pessoa);

    dao.setSessao(session);
    Pessoa ent = (Pessoa) dao.merge(this.pessoa);
    Assert.assertEquals(pessoa.getCpf(), ent.getCpf());
}
```

Fonte: (Os Autores, 2018).

5.3.1.2. Teste de Integração

O teste de integração analisa o funcionamento de partes diferentes do programa trabalhando em conjunto, então neste teste é chamado as funções e analisado se os dados e comportamentos obtidos são os mesmos que os esperados, como mostra a Figura 18.

Figura 18: Exemplo de Teste de Integração

```
@Test
public void salvarPessoaTesteIntegracao() throws ParseException {
    PessoaDAO dao = new PessoaDAO();

    Pessoa ent = (Pessoa) dao.merge(pessoa);

    Assert.assertEquals(pessoa.getCpf(), ent.getCpf());
}
```

Fonte: (Os Autores, 2018).

5.3.2. Teste de Aceitação

Os testes de aceitação têm como propósito validar as funcionalidades do *software*, utilizando-se assim as histórias de usuários como parâmetro de validação com seus critérios de aceitação.

Figura 19: Resultados dos Testes feitos no Cadastrar Disciplinas



Fonte: (Os Autores, 2018).

5.4. Implementação do Protótipo: dificuldades, facilidades e requisitos futuros

Com início de seu desenvolvimento no ano de 2017, o *software* passou por algumas modificações transformadoras durante todo o processo. Inicialmente foi pensado num projeto para abranger toda a faculdade e depois se expandir para as demais, mas orientaram-se para realizar apenas no curso de computação e posteriormente se propagar aos demais cursos, até porque o regimento de monitoria é variado de um curso para o outro. Seu desenvolvimento inicial teve como base um *layout* encontrado na internet, no qual foi pensado que supriria as

necessidades deste, no entanto, posteriormente foi visto que as tecnologias utilizadas naquele modelo não atenderiam mais ao *software*, foi então que decidiu-se reestruturar todo o projeto e criar um novo *layout* para o mesmo.

A equipe estava preocupada por mudar todo o *software* de repente, então determinou em dividir a equipe em duas partes, a *back-end* que trabalharia com a parte por trás do projeto e a *front-end* que seria responsável pelos *layouts* e documentações.

Para as novas adequações, foi necessário buscar tecnologias que atendessem o projeto até o final do desenvolvimento. Após decidi-las, vieram os principais problemas enfrentados pela equipe, pois com essas técnicas escolhidas para o projeto não haviam total dominância a respeito do conteúdo, então realizou-se uma reunião com a equipe para saber do conhecimento de cada membro. Visto que ninguém tinha entendimento das tecnologias, cada uma das partes do projeto focou nos estudos para poder aplica-las no desenvolvimento.

Devido às dificuldades encontradas neste semestre (2018/1) motivados pelas mudanças, houve um atraso no desenvolvimento do projeto. A equipe começou a aplicar o que foi adquirido com estudos, porém, ainda existia muitas dúvidas perante a tudo. No final do semestre foi onde tudo começou a encaminhar, pois a equipe possuía um pouco de conhecimento e já conseguia aplica-lo, então começaram a criar os protótipos base das telas existentes e as implementações das regras de negócios.

No semestre seguinte (2018/2), foram desenvolvidas algumas das aplicações existente no projeto como autenticação de usuário, gerenciamento de usuário, gerenciamento de monitorias, gerenciamento de disciplinas e gerenciamento de avisos, sendo essas as aplicações existentes até o real momento deste como podem ser visualizados no Apêndice A.

Após a realização desses artefatos, a equipe de *back-end* focou nos desenvolvimentos dos testes a serem realizados, que são os testes de integração e unitário. Obteve algumas dificuldades no início do desenvolvimento deles, mas com um pouco de prática foram realizados com sucesso.

Dentro de um futuro próximo é pretendido desenvolver as demais funcionalidades como o gerenciamento do fórum, edital de monitoria, emissão dos certificados, atribuição de acesso aos demais tipos de usuários, frequências dos usuários e emissão das fichas de avaliação. Após a finalização do projeto, poderão ser feitas atualizações e modificações para abranger outros cursos da instituição e, caso haja sucesso, poderá ser comercializado com outras instituições de ensino.

6. CONSIDERAÇÕES FINAIS

Durante o processo de desenvolvimento do projeto, a equipe pôde conhecer novas tecnologias necessárias para adquirir um maior conhecimento, além de explorar a cada uma delas para a entrega do protótipo.

O principal objetivo de desenvolver um *software* utilizando a metodologia ágil XP, ocorreu em diversas perspectivas da construção do projeto. As etapas alcançadas para este feito foram concluídas, apesar de alguns resultados não esperados, mas que não comprometeram o projeto em si.

Atrasos ocasionados pelas mudanças frequentes impediram o completo desenvolvimento de todas as funcionalidades das Histórias de Usuários especificadas no Apêndice A, postergando sua aplicação no semestre 2019/01 e dando assim continuidade no projeto durante os períodos seguintes.

As boas práticas trabalhadas durante o desenvolvimento, contribuíram tanto na melhora da aplicabilidade do projeto quanto na performance do trabalho em equipe. O XP foi de suma relevância para o *software*, foi através das práticas que decidiu-se trabalhar com *releases* curtas e sempre interagindo com cliente sobre os resultados obtidos, além de passar o conhecimento de cada um da equipe para todos.

No início a equipe teve bastante dificuldades para trabalhar com as novas tecnologias aderidas para o *software*, mas com estudos e comunicação entre os integrantes, foi adquirido uma maior maturidade e um conhecimento aprofundado fazendo com que o projeto caminhasse.

No ambiente de desenvolvimento adotado pela equipe, o XP apresentou algumas características diferentes do atual momento do projeto. As reuniões diárias tinham o objetivo de discutir tudo sobre projeto destacando pontos positivos e negativos, conseqüentemente eram destacadas soluções para problemas existentes ou realizava um estudo para o próximo encontro chegando a uma solução.

É compreendido a necessidade do processo utilizado ser incrementado no desenvolvimento, para ter melhor aproveitamento dos indivíduos e conhecimento de alterações nos cenários quando necessário.

Portanto, com a execução deste projeto, é proposto como trabalho futuro a inserção de novas funcionalidades e o melhoramento da qualidade tendo como base os resultados obtidos, facilitando cada vez mais a iteração do usuário com o *software*.

Com a execução dessa metodologia, percebe-se que o XP apresentou resultados positivos que, apesar de ser uma abordagem menos rigorosa que outras metodologias, cumpre-

se bem o seu objetivo que é proposto para equipes pequenas com um mínimo de documentação possível, cujo desenvolvimento acarreta mudanças constantes e o projeto tenha requisitos vagos.

O ambiente ágil trouxe nova experiência referente ao desenvolvimento de *software*, pois houve uma priorização de comunicação entre os envolvidos, gerando documentações para nortear a aplicação do processo. Até o momento, o XP conduziu o time a resultados satisfatórios, priorizando a entrega dos protótipos executável e documentados.

7. REFERÊNCIAS BIBLIOGRÁFICAS

ABREU, Thuany Oliveira et al. **A monitoria acadêmica na percepção dos graduandos de enfermagem**. 2014. 6 f. Artigo de Pesquisa (Graduação em Enfermagem) – UERJ, Revista Enfermagem, 2014. Disponível em: <https://www.researchgate.net/profile/Maria_Regina_Pimentel/publication/303910098_A_monitoria_academica_na_percepcao_dos_graduandos_de_enfermagem_Academic_monitoring_in_the_perception_of_undergraduate_nursing_students_La_monitoria_academica_en_la_percepcion_de_los_alumnos_de_pregra/links/575c3f2408aec91374abc3d8.pdf>. Acesso em: 22 maio 2018.

CUNHA, Dannilo M.; FILGUEIRA, Aquiles M.; VIANA, Gleiferson L. **ARQUITETURA DE SOFTWARE VOLTADA PARA A AVALIAÇÃO CONTÍNUA DO PROCESSO DE ENSINO-APRENDIZAGEM**. 2017. Disponível em: <<http://www2.ifrn.edu.br/ojs/index.php/HOLOS/article/view/5335>>. Acesso em: 28 set. 2018.

GIL, Antônio Carlos. **Como Elaborar Projetos de Pesquisa**. 6ª. ed. São Paulo: Atlas, 2018.

LARMAN, CRAIG. **UTILIZANDO UML E PADRÕES: Uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo**. 3ª. ed. Porto Alegre: Bookman, 2007. 695 p.

MANARA MILETTO, EVANDRO; CASTRO BERTAGNOLLI, SILVIA. **DESENVOLVIMENTO DE SOFTWARE II: INTRODUÇÃO AO DESENVOLVIMENTO WEB COM HTML, CSS, JAVASCRIPT E PHP**. Porto Alegre: Bookman, 2014. 266 p.

MATOSO, Leonardo Magela Lopes. **A IMPORTÂNCIA DA MONITORIA NA FORMAÇÃO ACADÊMICA DO MONITOR: UM RELATO DE EXPERIÊNCIA**. 2013. 7 f. Artigo de Pesquisa (Graduação em Enfermagem) - Universidade Potiguar-UNP, Catussaba, REVISTA CIENTÍFICA DA ESCOLA DA SAÚDE, 2013. Disponível em: <<https://repositorio.unp.br/index.php/catussaba/article/view/567>>. Acesso em: 21 maio 2018.

NATÁRIO, Elisete Gomes, SANTOS, Acácia Aparecida Angeli dos. **Programa de monitores para o ensino superior**. Estudos de Psicologia, 2010. Disponível em: <<http://www.redalyc.org/articulo.oa?id=395335744007>> ISSN 0103-166X. Acesso em: 24 de maio de 2018.

NUNES, João Batista Carvalho. **Monitoria Acadêmica: Espaço de Formação**. 2º Seminário de Iniciação à Docência da Universidade Federal do Rio Grande do Norte. 7 de outubro de 2005. Disponível em: <http://arquivos.info.ufrn.br/arquivos/20132161039fe41407857a2bf7803d137/Monitoria_4.pdf>. Acesso em 19 nov. 2017.

PEREIRA, Adriana Soares; SARTORI, André Paulo; KROTZ, Jacson Luis. **Sistema Especialista Web para Análise de Alimentos**. 2016. Acesso em: 20 maio 2018.

Presidência da República, Casa Civil Subchefia para Assuntos Jurídicos – Lei nº.9394/96. Disponível em: <http://www.planalto.gov.br/ccivil_03/leis/L9394compilado.htm>. Acesso em: 24 de maio de 2018.

PRESSMAN, Roger S. **Engenharia de Software: Uma Abordagem Profissional**. 7ª Ed. Porto Alegre: AMGC, 2011.

PRESSMAN, Roger S. **Engenharia de Software: Uma Abordagem Profissional**. 8ª Ed. Porto Alegre: AMGH, 2016.

PRIKLADNICKI, Rafael; WILLI, Renato; MILANI, Fabiano. **Métodos Ágeis para o Desenvolvimento de Software**. Porto Alegre: Bookman, 2014.

SBROCCO, José Henrique Teixeira de Carvalho; MACEDO, Paulo Cesar de. **Metodologias Ágeis: Engenharia de Software sob Medida**. 1 Ed. São Paulo: Érica, 2012.

SOMMERVILLE, Ian. **Engenharia de SOFTWARE**. 9ª Ed. São Paulo: Pearson, 2011.

TELES, Vinícius Manhães. **Extreme Programming: Aprenda como encarar seus usuários desenvolvendo software com agilidade e alta qualidade**. 2 Ed. São Paulo: Novatec, 2014.

TELES, Vinícius Manhães. **EXTREME PROGRAMMING**. Disponível em:
<<http://www.desenvolvimentoagil.com.br/xp/>>. Acesso em: 19 nov. 2017.

ANEXOS

ANEXO A - Edital N° 01/2018-1 Seleção de Monitores

A direção dos Bacharelados em Computação do Centro Universitário de Anápolis – UniEVANGÉLICA, torna pública as inscrições para o Processo de Seleção de Monitores, regendo-se está pelas regras e condições contidas no presente Edital, bem como no Regulamento de Monitoria.

Art. 1º. Dos objetivos do programa de monitoria:

- I. Estimular o interesse do aluno pela carreira docente, por meio de sua participação nas atividades de ensino do curso.
- II. Estimular a integração entre o corpo docente e discente, por meio da participação do aluno na vida acadêmica, especialmente no desenvolvimento de projetos de apoio à atividade docente, como busca incessante para melhoria do ensino em todos os níveis.
- III. Fornecer ao aluno ferramentas e condições para o aprofundamento técnico-científico voltado para docência.

Art. 2º. Dos deveres do monitor:

- I. Cumprir as atividades de monitoria com pontualidade e assiduidade.
- II. Planejar, em conjunto com o professor-orientador da disciplina, suas atividades de monitoria, elaborando assim o Plano de Trabalho semestral.
- III. Procurar, com frequência, o professor com a finalidade de receber orientações; apresentar as atividades e principais dúvidas sanadas durante a monitoria; auxiliar os alunos na realização de trabalhos teórico-práticos; esclarecer dúvidas sobre conteúdos estudados; apresentar a frequência dos discentes participantes; atender/apresentar outras solicitações do professor referentes à atividade de monitoria.
- IV. Assinar a sua ficha de frequência mensal das atividades de monitoria desempenhadas e entregar o boletim na Secretaria Setorial até o dia 05 de cada mês subsequente.

- V. Elaborar o Relatório Final sobre as atividades de monitoria desempenhadas ao longo do semestre, que deve ser entregue no final do semestre.
 - VI. É dever do monitor, informar à turma, em todos os meios de comunicação disponíveis (mural da sala, mural da secretaria, e-mail, grupos de discussão, entre outros) sobre datas, horários e demais notificações acerca das atividades de monitoria.
 - VII. Facilitar o relacionamento entre os alunos e professores, contribuindo para o cumprimento do plano de ensino da disciplina.
 - VIII. Apresentar ao professor orientador, quando for o caso, proposta de seu desligamento do Programa, com antecedência mínima de 15 dias.
- IX. Avaliar o andamento da disciplina do ponto de vista do aluno, apresentando sugestões ao professor.
- IX. É vedado aos monitores substituir o professor nas aulas de sua responsabilidade, aplicar provas e trabalhos de verificação do rendimento escolar, ministrar cursos de acompanhamento, em caráter particular, para alunos que estiverem cursando disciplinas objeto de Monitoria e exercer quaisquer atividades de caráter administrativo, de julgamento, de verificação de aprendizagem e de supervisão de estágio.

Art. 3º. Das vagas

- I. As disciplinas, professores responsáveis, modalidades da monitoria e o número de vagas estão disponíveis no Anexo I deste Edital.

Art.4º. Do período das inscrições

- I. O acadêmico deve procurar o docente e informar a sua intenção em ser monitor.
- II. O professor fará a sua indicação / inscrição, observando o calendário em anexo (ou, em qualquer tempo, ao longo do semestre, visando a melhoria do processo ensino-aprendizagem).
- III. O professor deverá utilizar o formulário online, disponível em <https://goo.gl/VJ4eY1>, para fazer a inscrição do acadêmico.

Art.5º. Da seleção

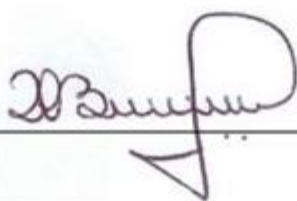
- I. Se dará por meio de indicação pelo docente.

- II. Se o número de monitores for maior que a quantidade de vagas, o critério de seleção será a média geral do acadêmico.
- III. Se aplicável, o professor poderá ampliar o número de vagas, considerando o tamanho da turma e/ou complexidade de conteúdo da disciplina.
- IV. As datas de cada etapa deste processo estão no Anexo II deste Edital.

Art.6º. Das exigências para candidatar-se à monitoria

- I. Ser estudante devidamente matriculado nos Cursos Superiores de Computação do Centro Universitário de Anápolis.
- II. Estar cursando ou ter cursado a disciplina no qual se deseja candidatar-se, com aproveitamento igual ou superior a sete (7,0).
 - a. No caso da monitoria ser no mesmo horário de aula da disciplina que o aluno se candidatou, será necessário o acompanhamento do professor da disciplina, fornecendo exercícios e o apoio necessário para o bom desempenho das atividades do monitor.
- III. Ter disponibilidade de 4 horas semanais, podendo ser presencial e/ou a distância. O horário da monitoria será definido em conjunto com os alunos, o monitor e o professor responsável pela disciplina.
- IV. Inscrever em, no máximo, três disciplinas para realizar as atividades como monitor.
- V. Não ter sido advertido em nenhuma das disciplinas.

Anápolis, 02 de fevereiro de 2018.



*Profª Ma. Viviane C. B. Pocivi, CSM, CSPO
UniEvangélica - Cursos de Computação
Diretora, Port. 4 de 25/01/2010*

APÊNDICES

APÊNDICE A – Histórias de Usuários ¹

[UC001] Autenticar Usuário

Usuários/Perfis

Aluno, monitor, professor e administrador.

Pré-Condições

- O usuário deverá acessar o portal.
- O usuário deverá estar cadastrado na base de dados.

Como Acessar

Acessar o portal > [UC001] Autenticar Usuário

História de Usuário

COMO pessoa
QUERO me autenticar
PARA acessar o sistema.

Cenários

Código	Descrição	Tipo	Cenário	Referência
CN-01	Autenticar	Autenticação	DADO QUE uma pessoa deseja acessar o sistema, QUANDO entra no site, E preenche os campos do formulário E clica em “Login”, ENTÃO ela é autenticada no sistema.	
CN-02	Campo obrigatório não informado	Erro	DADO QUE a pessoa não preencheu um campo obrigatório em um formulário, QUANDO clica em “Login”, ENTÃO visualiza a mensagem de erro: “Matrícula/Senha é obrigatório”.	
CN-03	Falha com banco de dados	Erro	DADO QUE a pessoa solicita uma transação com o banco de dados, QUANDO ela confirma a transação E o banco de dados falha ao executar, ENTÃO ela visualiza a mensagem de falha de conexão com o banco de dados: “Não foi possível conectar com o banco de dados”.	
CN-04	Usuário não encontrado	Erro	DADO QUE a pessoa insere um usuário inválido em um formulário, QUANDO clica em “Login”	

¹ Por estar utilizando uma metodologia iterativa e incremental, esta é a 5ª versão.

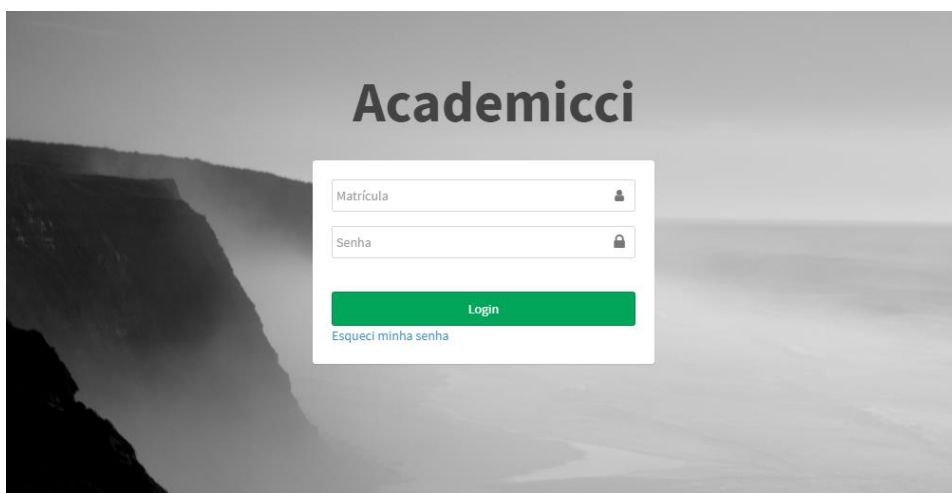
			ENTÃO visualiza a mensagem de erro: “Matrícula ou Senha incorretos”.	
CN-05	Redefinir senha	Redefinição de senha	DADO QUE uma pessoa não lembre a senha, QUANDO clica em Esqueci minha senha, ENTÃO é redirecionada para a redefinição da senha.	

Critérios de Aceitação

Código	Descrição	Aceito?

Protótipos

P1 – Realizar Login



Descrição dos Campos

Nome	Descrição	Tipo	Tipo de Dado (tamanho)	Editável	Obrigatório	Regras
Matrícula	Campo para inserir a matrícula	Input text	Numérico (7)	Sim	Sim	N/A
Senha	Campo para inserir a senha	Password	Alfanumérico (32)	Sim	Sim	O campo deverá possuir máscara para senha.

Descrição dos Comandos

Nome	Ação	Tipo	Restrições
Acesso	O sistema exibirá o formulário de login.	Link	N/A
Esqueci minha senha	O usuário será redirecionado para página de recuperação de senha.	Link	N/A

Login	O usuário será redirecionado para a página inicial do sistema.	Botão	O botão será clicável.
-------	--	-------	------------------------

Regras de Apresentação

Descrição
<ul style="list-style-type: none"> N/A

[UC002] Cadastrar Pessoa

Usuários/Perfis

Aluno, monitor, professor e administrador.

Pré-Condições

- O usuário deverá estar logado.
- O usuário deverá ser administrador.

Como Acessar

Acessar o portal > Realizar Login > Usuários > [UC002] Cadastrar Pessoa

História de Usuário

COMO administrado

QUERO cadastrar usuários

PARA acessarem o sistema.

Cenários

Código	Descrição	Tipo	Cenário	Referência
CN-01	Cadastrar pessoa	Cadastramento	DADO QUE é administrador do sistema, QUANDO clica em “Cadastrar Pessoa”, E preenche os campos do formulário, E clica em “Salva”, ENTÃO visualiza a mensagem de: “Pessoa Salva com sucesso”.	
CN-02	Campo obrigatório não informado	Erro	DADO QUE a pessoa não preencheu um campo obrigatório em um formulário, QUANDO clica em Salvar, ENTÃO visualiza a mensagem de erro: “Campo “tal” é obrigatório”.	

Critérios de Aceitação

Código	Descrição	Aceito?

Protótipos

P1 – Cadastrar Pessoa

Cadastrar Pessoa

Matrícula:

Nome:

Sexo:

Data de Nascimento:

E-mail:

CPF:

RG:

DDD:

Número:

Senha:

Ativo:

Descrição dos Campos

Nome	Descrição	Tipo	Tipo de Dado (tamanho)	Editável	Obrigatório	Regras
Matrícula	Campo para inserir a matrícula	Input text	Numérico (7)	Sim	Sim	N/A
Nome	Campo para inserir o Nome	Input text	Texto (100)	Sim	Sim	N/A
Sexo	Campo para inserir o sexo	Selection	Seleção	Não	Sim	O campo terá definido o sexo, usuário só classificará qual.
Data de Nascimento	Campo para inserir a data de nascimento	Input text Selection	Alfanumérico (10)	Sim	Sim	A data deve ser preenchida sem espaçamento
E-mail	Campo para inserir o e-mail	Input text	Alfanumérico (100)	Sim	Sim	N/A
CPF	Campo para inserir o CPF	Input text	Numérico (100)	Sim	Sim	N/A
RG	Campo para inserir o RG	Input text	Numérico (100)	Sim	Sim	N/A

DDD	Campo para inserir o DDD	Input text	Numérico (2)	Sim	Sim	N/A
Número	Campo para inserir o número do telefone	Input text	Numérico (9)	Sim	Sim	N/A
Senha	Campo para inserir a senha	Password	Alfanumérico (32)	Sim	Sim	O campo deverá possuir máscara para senha.
Ativo	Campo para inserir se está ativo ou não	Selection	Seleção	Não	Não	N/A

Descrição dos Comandos

Nome	Ação	Tipo	Restrições
Salvar	A pessoa será cadastrada.	Botão	O botão será clicável
Fechar	O usuário será redirecionado para página Usuários.	Botão	O botão será clicável

Regras de Apresentação

Descrição
<ul style="list-style-type: none"> N/A

[UC003] Editar Pessoa

Usuários/Perfis

Aluno, monitor, professor e administrador.

Pré-Condições

- O usuário deverá estar logado.
- O usuário deverá estar cadastrado.

Como Acessar

Acessar o portal > Realizar Login > Usuários > Ações > [UC003] Editar Pessoa

História de Usuário

COMO pessoa

QUERO editar dados cadastrados

PARA manter os dados sempre atualizados.

Cenários

Código	Descrição	Tipo	Cenário	Referência
--------	-----------	------	---------	------------

CN-01	Editar pessoa	Edição	DADO QUE é usuário do sistema, QUANDO clica no ícone de editar, E edita os campos necessários do formulário, E clica em “Salvar”, ENTÃO visualiza a mensagem de: “Pessoa Salva com sucesso”.
CN-02	Campo obrigatório não informado	Erro	DADO QUE a pessoa não preencheu um campo obrigatório em um formulário, QUANDO clica em “Salvar”, ENTÃO visualiza a mensagem de erro: “Campo “tal” é obrigatório”.

Critérios de Aceitação

Código	Descrição	Aceito?

Protótipos


P1 – Editar Pessoa

Cadastrar Pessoa

Matrícula:

Nome:

Sexo: ▼

Data de Nascimento: 

E-mail:

CPF:

RG:

DDD:

Número:

Senha:

Ativo:

Descrição dos Campos

Nome	Descrição	Tipo	Tipo de Dado (tamanho)	Editável	Obrigatório	Regras
Matrícula	Campo para inserir a matrícula	Input text	Numérico (7)	Sim	Sim	N/A
Nome	Campo para inserir o Nome	Input text	Texto (100)	Sim	Sim	N/A
Sexo	Campo para inserir o sexo	Selection	Seleção	Não	Sim	O campo terá definido o sexo, usuário só classificará qual.
Data de Nascimento	Campo para inserir a data de nascimento	Input text Selection	Alfanumérico (10)	Sim	Sim	A data deve ser preenchida sem espaçamento
E-mail	Campo para inserir o e-mail	Input text	Alfanumérico (100)	Sim	Sim	N/A
CPF	Campo para inserir o CPF	Input text	Numérico (100)	Sim	Sim	N/A
RG	Campo para inserir o RG	Input text	Numérico (100)	Sim	Sim	N/A
DDD	Campo para inserir o DDD	Input text	Numérico (2)	Sim	Sim	N/A
Número	Campo para inserir o número do telefone	Input text	Numérico (9)	Sim	Sim	N/A
Senha	Campo para inserir a senha	Password	Alfanumérico (32)	Sim	Sim	O campo deverá possuir máscara para senha.
Ativo	Campo para inserir se está ativo ou não	Selection	Seleção	Não	Não	N/A

Descrição dos Comandos

Nome	Ação	Tipo	Restrições
Salvar	A pessoa será cadastrada.	Botão	O botão será clicável
Fechar	O usuário será redirecionado para página Usuários.	Botão	O botão será clicável

Regras de Apresentação

Descrição
<ul style="list-style-type: none">N/A

[UC004] Excluir Pessoa

Usuários/Perfis

Aluno, monitor, professor e administrador.

Pré-Condições

- O usuário deverá estar logado.
- O usuário deverá estar cadastrado.
-

Como Acessar

Acessar o portal > Realizar Login > Usuários > Ações > [UC004] Excluir Pessoa

História de Usuário

COMO administrador

QUERO excluir pessoas cadastradas

PARA não acessar mais o sistema.

Cenários

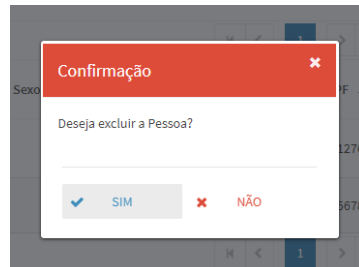
Código	Descrição	Tipo	Cenário	Referência
CN-01	Excluir pessoa	Excluir	DADO QUE sou administrador do sistema, QUANDO clica no ícone de excluir de uma pessoa, E clica em “SIM”; ENTÃO visualiza a mensagem de: “Pessoa Excluída com sucesso”.	

Critérios de Aceitação

Código	Descrição	Aceito?

Protótipos

P1 – Excluir Pessoa



Descrição dos Campos

Nome	Descrição	Tipo	Tipo de Dado (tamanho)	Editável	Obrigatório	Regras
Confirmação	Input text	Text	Texto	Não	Não	Apenas uma mensagem para confirmação da exclusão

Descrição dos Comandos

Nome	Ação	Tipo	Restrições
SIM	A pessoa será excluída do sistema.	Botão	O botão será clicável
Não	Sairá da página de exclusão.	Botão	O botão será clicável

Regras de Apresentação

Descrição
<ul style="list-style-type: none">N/A

[UC005] Listar Pessoa

Usuários/Perfis

Aluno, monitor, professor e administrador.

Pré-Condições

- O usuário deverá estar logado.

Como Acessar

Acessar o portal > Realizar Login > Usuários > [UC005] Listar Pessoa

História de Usuário

COMO professor ou administrador

QUERO listar usuários

PARA visualizar todos usuários cadastrados.

Cenários

Código	Descrição	Tipo	Cenário	Referência
CN-01	Listar pessoa	Listagem	DADO QUE sou professor ou administrador do sistema, QUANDO clica em “Usuários”, ENTÃO visualiza todos usuários cadastrados.	
CN-02	Sem pessoa cadastrada	erro	DADO QUE sou professor ou administrador do sistema, QUANDO clica em “Usuários”, ENTÃO visualiza a mensagem: “Nenhuma pessoa cadastrada”.	

CrITÉrios de Aceitação

Código	Descrição	Aceito?

Protótipos

P1 – Listar Pessoa

Descrição dos Campos

Nome	Descrição	Tipo	Tipo de Dado (tamanho)	Editável	Obrigatório	Regras
Matrícula	Campo mostra a matrícula	Input text	Numérico	Não	Sim	Somente visualização dos

						dados cadastrados.
Nome	Campo mostra o Nome	Input text	Texto	Não	Sim	Somente visualização dos dados cadastrados.
Sexo	Campo mostra o sexo	Input text	Texto	Não	Sim	Somente visualização dos dados cadastrados.
Data de Nascimento	Campo mostra a data de nascimento	Input text Selection	Alfanumérico	Não	Sim	Somente visualização dos dados cadastrados.
E-mail	Campo mostra o e-mail	Input text	Alfanumérico	Não	Sim	Somente visualização dos dados cadastrados.
CPF	Campo mostra o CPF	Input text	Numérico	Não	Sim	Somente visualização dos dados cadastrados.
RG	Campo mostra o RG	Input text	Numérico (100)	Não	Sim	Somente visualização dos dados cadastrados.
Número	Campo mostra o número do telefone	Input text	Numérico (9)	Não	Sim	Somente visualização dos dados cadastrados.
Status	Campo mostra se está ativo ou não	Selection	Seleção	Não	Sim	Somente visualização dos dados cadastrados.
Opções	Campo mostra as ações permitidas	Selection	seleção	Não	Sim	

Descrição dos Comandos

Nome	Ação	Tipo	Restrições

Regras de Apresentação

Descrição
<ul style="list-style-type: none">N/A

[UC006] Cadastrar Disciplina

Usuários/Perfis

Aluno, monitor, professor e administrador.

Pré-Condições

- O usuário deverá estar logado.
- O usuário deverá ser administrador.

Como Acessar

Acessar o portal > Realizar Login > Disciplinas > [UC006] Cadastrar Disciplina

História de Usuário

COMO administrador

QUERO cadastrar disciplinas

PARA manter o cadastro no sistema.

Cenários

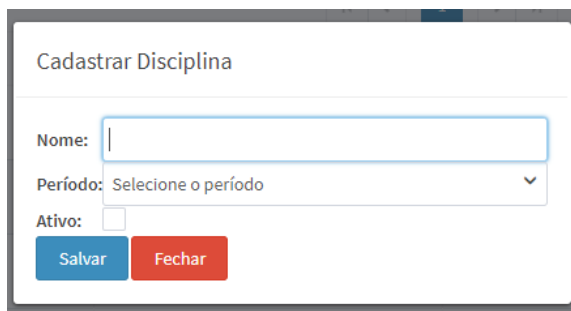
Código	Descrição	Tipo	Cenário	Referência
CN-01	Cadastrar disciplina	Cadastramento	DADO QUE é administrador do sistema, QUANDO clica em “Cadastrar Disciplina”, E preenche os campos do formulário, E clica em “Salvar”, ENTÃO visualiza a mensagem de: “Disciplina Salva com sucesso”.	
CN-02	Campo obrigatório não informado	Erro	DADO QUE a pessoa não preencheu um campo obrigatório em um formulário, QUANDO clica em “Salvar”, ENTÃO visualiza a mensagem de erro: “Campo “tal” é obrigatório”.	

Critérios de Aceitação

Código	Descrição	Aceito?

Protótipos

P1 – Cadastrar Disciplina



Descrição dos Campos

Nome	Descrição	Tipo	Tipo de Dado (tamanho)	Editável	Obrigatório	Regras
Nome	Campo para inserir o Nome	Input text	Texto (100)	Sim	Sim	N/A
Período	Campo para inserir o período	selection	Seleção	Não	Sim	N/A
Ativo	Campo para inserir se está ativo ou não	Selection	Seleção	Não	Não	N/A

Descrição dos Comandos

Nome	Ação	Tipo	Restrições
Salvar	A disciplina será cadastrada.	Botão	O botão será clicável
Fechar	O usuário será redirecionado para página Disciplinas.	Botão	O botão será clicável

Regras de Apresentação

Descrição
<ul style="list-style-type: none">N/A

[UC007] Editar Disciplina

Usuários/Perfis

Aluno, monitor, professor e administrador.

Pré-Condições

- O usuário deverá estar logado.
- O usuário deverá estar cadastrado.

Como Acessar

Acessar o portal > Realizar Login > Disciplinas > Ações > [UC007] Editar Disciplina

História de Usuário

COMO administrador

QUERO editar disciplinas cadastradas

PARA manter os dados sempre atualizados.

Cenários

Código	Descrição	Tipo	Cenário	Referência
CN-01	Editar disciplina	Edição	DADO QUE é administrador do sistema, QUANDO clica no ícone de editar, E edita os campos necessários do formulário, E clica em “Salvar”, ENTÃO visualiza a mensagem de: “Disciplina Salva com sucesso”.	
CN-02	Campo obrigatório não informado	Erro	DADO QUE a pessoa não preencheu um campo obrigatório em um formulário, QUANDO clica em “Salvar”, ENTÃO visualiza a mensagem de erro: “Campo “tal” é obrigatório”.	

Critérios de Aceitação

Código	Descrição	Aceito?

Protótipos

P1 – Editar Disciplina

Cadastrar Disciplina

Nome:

Período:

Ativo:

Descrição dos Campos

Nome	Descrição	Tipo	Tipo de Dado (tamanho)	Editável	Obrigatório	Regras
------	-----------	------	------------------------	----------	-------------	--------

Nome	Campo para inserir o Nome	Input text	Texto (100)	Sim	Sim	N/A
Período	Campo para inserir o período	selection	Seleção	Não	Sim	N/A
Ativo	Campo para inserir se está ativo ou não	Selection	Seleção	Não	Não	N/A

Descrição dos Comandos

Nome	Ação	Tipo	Restrições
Salvar	A disciplina será editada.	Botão	O botão será clicável
Fechar	O usuário será redirecionado para página Usuários.	Botão	O botão será clicável

Regras de Apresentação

Descrição
<ul style="list-style-type: none"> N/A

[UC008] Excluir Disciplina

Usuários/Perfis

Aluno, monitor, professor e administrador.

Pré-Condições

- O usuário deverá estar logado.
- O usuário deverá estar cadastrado.

Como Acessar

Acessar o portal > Realizar Login > Disciplinas > Ações > [UC008] Excluir Disciplina

História de Usuário

COMO administrador

QUERO excluir disciplinas cadastradas

PARA não terem mais acesso a ela.

Cenários

Código	Descrição	Tipo	Cenário	Referência
--------	-----------	------	---------	------------

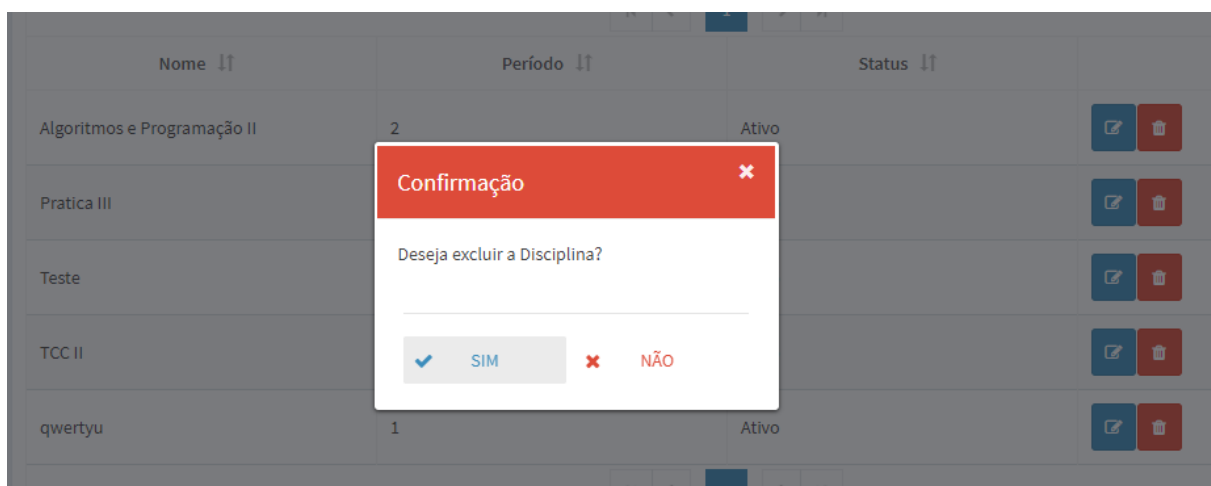
CN-01	Excluir pessoa	Excluir	DADO QUE é administrador do sistema, QUANDO clica no ícone de excluir de uma disciplina, E clica em “SIM”; ENTÃO visualiza a mensagem de: “Disciplina Excluída com sucesso”.	
-------	----------------	---------	---	--

Critérios de Aceitação

Código	Descrição	Aceito?

Protótipos

P1 – Excluir Disciplina



Descrição dos Campos

Nome	Descrição	Tipo	Tipo de Dado (tamanho)	Editável	Obrigatório	Regras
Confirmação	Input text	Text	Texto	Não	Sim	Apenas uma mensagem para confirmação da exclusão

Descrição dos Comandos

Nome	Ação	Tipo	Restrições
SIM	A disciplina será excluída do sistema.	Botão	O botão será clicável
Não	Sairá da página de exclusão.	Botão	O botão será clicável

Regras de Apresentação

Descrição
<ul style="list-style-type: none"> N/A

[UC009] Listar Disciplina

Usuários/Perfis

Aluno, monitor, professor e administrador.

Pré-Condições

- O usuário deverá estar logado.
- O usuário deverá ser administrador.

Como Acessar

Acessar o portal > Realizar Login > Disciplinas > [UC009] Listar Disciplina

História de Usuário

COMO administrador

QUERO listar disciplinas

PARA visualizar todas disciplinas cadastradas.

Cenários

Código	Descrição	Tipo	Cenário	Referência
CN-01	Listar disciplina	Listagem	DADO QUE é administrador do sistema, QUANDO clica em “Disciplinas”, ENTÃO visualiza todas disciplinas cadastradas.	
CN-02	Sem disciplina cadastrada	erro	DADO QUE é administrador do sistema, QUANDO clica em “Disciplinas”, ENTÃO visualiza a mensagem: “Nenhuma disciplina cadastrada”.	

Critérios de Aceitação

Código	Descrição	Aceito?

Protótipos

P1 – Listar Pessoa

Nome	Período	Status	Opções
Algoritmos e Programação II	2	Ativo	
Pratica III	10	Ativo	
Teste	1	Ativo	
TCC II	10	Ativo	
qwertyu	1	Ativo	

Descrição dos Campos

Nome	Descrição	Tipo	Tipo de Dado (tamanho)	Editável	Obrigatório	Regras
Nome	Campo mostra o Nome	Input text	Texto	Não	Sim	Somente visualização dos dados cadastrados.
Período	Campo mostra o período	Selection	Seleção	Não	Sim	
Status	Campo mostra se está ativo ou não	Selection	Seleção	Não	Sim	Somente visualização dos dados cadastrados.
Opções	Campo mostra as ações permitidas	Selection	seleção	Não	Sim	N/A

Descrição dos Comandos

Nome	Ação	Tipo	Restrições

Regras de Apresentação

Descrição
<ul style="list-style-type: none"> N/A

[UC010] Cadastrar Monitoria

Usuários/Perfis

Aluno, monitor, professor e administrador.

Pré-Condições

- O usuário deverá estar logado.
- O usuário deverá ser administrador.

Como Acessar

Acessar o portal > Realizar Login > Monitorias > [UC010] Cadastrar Monitoria

História de Usuário

COMO administrador

QUERO cadastrar monitorias

PARA disponibilizar as monitorias aos usuários.

Cenários

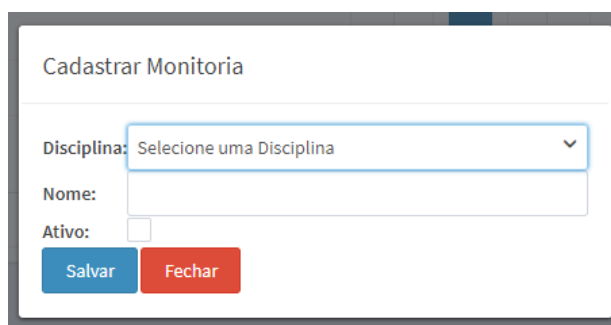
Código	Descrição	Tipo	Cenário	Referência
CN-01	Cadastrar monitoria	Cadastramento	DADO QUE é administrador do sistema, QUANDO clica em “Cadastrar Monitoria”, E preenche os campos do formulário, E clica em “Salvar”, ENTÃO visualiza a mensagem de: “Monitoria Salva com sucesso”.	
CN-02	Campo obrigatório não informado	Erro	DADO QUE a pessoa não preencheu um campo obrigatório em um formulário, QUANDO clica em “Salvar”, ENTÃO visualiza a mensagem de erro: “Campo “tal” é obrigatório”.	

Critérios de Aceitação

Código	Descrição	Aceito?

Protótipos

P1 – Cadastrar Monitoria



Descrição dos Campos

Nome	Descrição	Tipo	Tipo de Dado (tamanho)	Editável	Obrigatório	Regras
Disciplina	Campo para inserir a disciplina	selection	Seleção	Não	Sim	N/A
Nome	Campo para inserir o Nome	Input text	Texto (100)	Sim	Sim	N/A
Ativo	Campo para inserir se está ativo ou não	Selection	Seleção	Não	Não	N/A

Descrição dos Comandos

Nome	Ação	Tipo	Restrições
Salvar	A monitoria será cadastrada.	Botão	O botão será clicável
Fechar	O usuário será redirecionado para página Monitorias.	Botão	O botão será clicável

Regras de Apresentação

Descrição
<ul style="list-style-type: none"> N/A

[UC011] Editar Monitoria

Usuários/Perfis

Aluno, monitor, professor e administrador.

Pré-Condições

- O usuário deverá estar logado.
- O usuário deverá estar cadastrado.

Como Acessar

Acessar o portal > Realizar Login > Monitorias > Ações > [UC011] Editar Monitoria

História de Usuário

COMO administrador

QUERO editar monitorias cadastradas

PARA manter os dados atualizados.

Cenários

Código	Descrição	Tipo	Cenário	Referência
CN-01	Editar monitoria	Edição	DADO QUE é administrador do sistema, QUANDO clica no ícone de editar, E edita os campos necessários do formulário, E clica em “Salvar”, ENTÃO visualiza a mensagem de: “Monitoria Salva com sucesso”.	
CN-02	Campo obrigatório não informado	Erro	DADO QUE a pessoa não preencheu um campo obrigatório em um formulário, QUANDO clica em “Salvar”, ENTÃO visualiza a mensagem de erro: “Campo “tal” é obrigatório”.	

Critérios de Aceitação

Código	Descrição	Aceito?

Protótipos

P1 – Editar Monitoria

O protótipo mostra uma interface de usuário para 'Cadastrar Monitoria'. O formulário contém os seguintes campos e controles:

- Disciplina:** Um campo de texto com o valor 'qwertyu' e uma seta para baixo, indicando uma lista suspensa.
- Nome:** Um campo de texto com o valor 'Monitoria de qwertyu'.
- Ativo:** Um campo de texto com um ícone de caixa de seleção marcada.
- Botões:** Dois botões de ação: 'Salvar' (em azul) e 'Fechar' (em vermelho).

Descrição dos Campos

Nome	Descrição	Tipo	Tipo de Dado (tamanho)	Editável	Obrigatório	Regras
Disciplina	Campo para inserir a disciplina	selection	Seleção	Não	Sim	N/A
Nome	Campo para inserir o Nome	Input text	Texto (100)	Sim	Sim	N/A
Ativo	Campo para inserir se está ativo ou não	Selection	Seleção	Não	Não	N/A

Descrição dos Comandos

Nome	Ação	Tipo	Restrições
Salvar	A monitoria será editada.	Botão	O botão será clicável
Fechar	O usuário será redirecionado para página Monitorias.	Botão	O botão será clicável

Regras de Apresentação

Descrição
<ul style="list-style-type: none">N/A

[UC012] Excluir Monitoria

Usuários/Perfis

Aluno, monitor, professor e administrador.

Pré-Condições

- O usuário deverá estar logado.
- O usuário deverá estar cadastrado.

Como Acessar

Acessar o portal > Realizar Login > Monitorias > Ações > [UC012] Excluir Monitoria

História de Usuário

COMO administrador

QUERO excluir monitorias cadastradas

PARA não terem mais acesso a ela.

Cenários

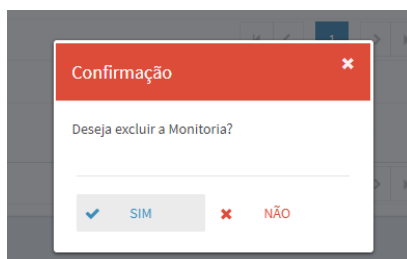
Código	Descrição	Tipo	Cenário	Referência
CN-01	Excluir monitoria	Excluir	DADO QUE é administrador do sistema, QUANDO clica no ícone de excluir de uma monitoria, E clica em “SIM”; ENTÃO visualiza a mensagem de: “Monitoria Excluída com sucesso”.	

Critérios de Aceitação

Código	Descrição	Aceito?

Protótipos

P1 – Excluir Monitoria



Descrição dos Campos

Nome	Descrição	Tipo	Tipo de Dado (tamanho)	Editável	Obrigatório	Regras
Confirmação	Input text	Text	Texto	Não	Sim	Apenas uma mensagem para confirmação da exclusão

Descrição dos Comandos

Nome	Ação	Tipo	Restrições
SIM	A monitoria será excluída do sistema.	Botão	O botão será clicável
Não	Sairá da página de exclusão.	Botão	O botão será clicável

Regras de Apresentação

Descrição
<ul style="list-style-type: none">N/A

[UC013] Listar Monitorias

Usuários/Perfis

Aluno, monitor, professor e administrador.

Pré-Condições

- O usuário deverá estar logado.

Como Acessar

Acessar o portal > Realizar Login > Monitorias > [UC013] Listar Monitorias

História de Usuário

COMO pessoa

QUERO listar monitorias

PARA visualizar todas monitorias cadastradas.

Cenários

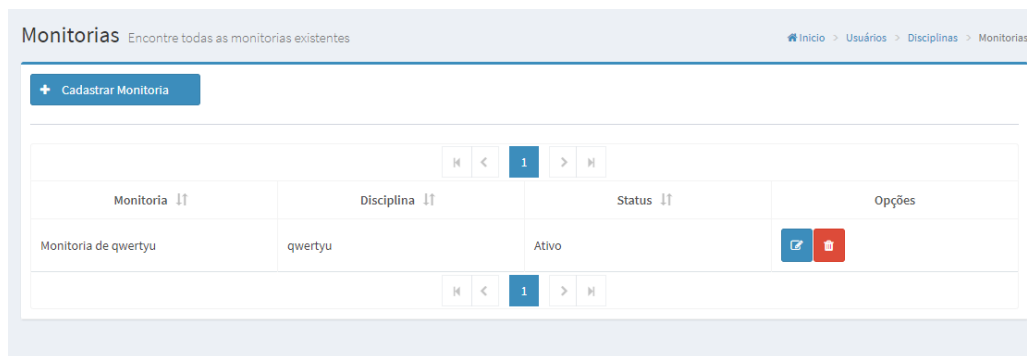
Código	Descrição	Tipo	Cenário	Referência
CN-01	Listar monitorias	Listagem	DADO QUE é usuário do sistema, QUANDO clica em “Monitorias”, ENTÃO visualiza todas monitorias cadastradas.	
CN-02	Sem monitoria cadastrada	Erro	DADO QUE é usuário do sistema, QUANDO clica em “Monitorias”, ENTÃO visualiza a mensagem: “Nenhuma Monitoria cadastrada”.	

Critérios de Aceitação

Código	Descrição	Aceito?

Protótipos

P1 – Listar Monitorias



Descrição dos Campos

Nome	Descrição	Tipo	Tipo de Dado (tamanho)	Editável	Obrigatório	Regras
Monitoria	Campo mostra o nome da monitoria	Text	Texto	Não	Sim	Somente visualização dos dados cadastrados.
Disciplina	Campo mostra a disciplina	Text	Texto	Não	Sim	
Status	Campo mostra se está ativo ou não	Text	Texto	Não	Sim	Somente visualização dos dados cadastrados.
Opções	Campo mostra as ações permitidas	Selection	seleção	Não	Sim	N/A

Descrição dos Comandos

Nome	Ação	Tipo	Restrições

Regras de Apresentação

Descrição
<ul style="list-style-type: none"> N/A

[UC014] Cadastrar aviso

Usuários/Perfis

Aluno, monitor, professor e administrador.

Pré-Condições

- O usuário deverá estar logado.

Como Acessar

Acessar o portal > Realizar Login > Avisos > [UC014] Cadastrar aviso

História de Usuário

COMO monitor, professor e administrador

QUERO cadastrar avisos

PARA enviar aos destinatários.

Cenários

Código	Descrição	Tipo	Cenário	Referência
CN-01	Cadastrar aviso	Listagem	DADO QUE é usuário específicos do sistema, QUANDO clica em “Criar aviso”, E preenche os campos do formulário, E clica em “Salvar”, ENTÃO visualiza a mensagem de “Aviso salvo com sucesso”.	
CN-02	Campo obrigatório não informado	Erro	DADO QUE a pessoa não preencheu um campo obrigatório em um formulário, QUANDO clica em “Salvar”, ENTÃO visualiza a mensagem de “Campo “tal” é obrigatório”.	

Critérios de Aceitação

Código	Descrição	Aceito?

Protótipos

P1 – Cadastrar Aviso

Cadastrar Aviso

Autor

Monitoria

Descrição

Assunto

Data

Descrição dos Campos

Nome	Descrição	Tipo	Tipo de Dado (tamanho)	Editável	Obrigatório	Regras
Autor	Campo para inserir autor	Selection	Seleção	Não	Sim	Classificar o autor.
Monitoria	Campo para inserir monitoria	Selection	Seleção	Não	Sim	Classificar monitoria.
Descrição	Campo para inserir a descrição do aviso	Input text	Alfanumérico(100)	Sim	Sim	N/A
Assunto	Campo para inserir o assunto do aviso	Input text	Alfanumérico(500)	Sim	Sim	N/A
Data	Campo para inserir a data	Input text Selection	Alfanumérico (10)	Sim	Sim	A data deve ser preenchida sem espaçamento

Descrição dos Comandos

Nome	Ação	Tipo	Restrições
Salvar	O aviso será enviado para a pessoa.	Botão	Botão clicável
Fechar	O usuário será redirecionado para página Avisos.	Botão	Botão clicável

Regras de Apresentação

Descrição
<ul style="list-style-type: none"> N/A

[UC015] Editar aviso

Usuários/Perfis

Aluno, monitor, professor e administrador.

Pré-Condições

- O usuário deverá estar logado.

Como Acessar

Acessar o portal > Realizar Login > Avisos > Ações > [UC015] Editar aviso

História de Usuário

COMO monitor, professor e administrador

QUERO editar avisos
PARA fazer alterações quando necessário.

Cenários

Código	Descrição	Tipo	Cenário	Referência
CN-01	Editar aviso	Edição	DADO QUE é usuário específicos do sistema, QUANDO clica no ícone de editar, E edita os campos necessários do formulário, E clica em “Salvar”, ENTÃO visualiza a mensagem de “Aviso salvo com sucesso”.	
CN-02	Campo obrigatório não informado	Erro	DADO QUE a pessoa não preencheu um campo obrigatório em um formulário, QUANDO clica em “Salvar”, ENTÃO visualiza a mensagem de “Campo “tal” é obrigatório”.	

Critérios de Aceitação

Código	Descrição	Aceito?

Protótipos P1 – Editar Aviso

Cadastrar Aviso

Autor Vinicius Mateus Martins ▼

Monitoria TCC II - 2018/02 ▼

Descrição qwerty

Assunto wert

Data 03-12-2018

Salvar
Fechar

Descrição dos Campos

Nome	Descrição	Tipo	Tipo de Dado (tamanho)	Editável	Obrigatório	Regras
Autor	Campo para inserir autor	Selection	Seleção	Não	Sim	Classificar o autor.
Monitoria	Campo para inserir monitoria	Selection	Seleção	Não	Sim	Classificar monitoria.
Descrição	Campo para inserir a descrição do aviso	Input text	Alfanumérico(100)	Sim	Sim	N/A
Assunto	Campo para inserir o assunto do aviso	Input text	Alfanumérico(500)	Sim	Sim	N/A
Data	Campo para inserir a data	Input text Selection	Alfanumérico (10)	Sim	Sim	A data deve ser preenchida sem espaçamento

Descrição dos Comandos

Nome	Ação	Tipo	Restrições
Salvar	O aviso será editado para a pessoa.	Botão	Botão clicável
Fechar	O usuário será redirecionado para página Avisos.	Botão	Botão clicável

Regras de Apresentação

Descrição
<ul style="list-style-type: none">N/A

[UC016] Excluir Aviso

Usuários/Perfis

Aluno, monitor, professor e administrador.

Pré-Condições

- O usuário deverá estar logado.
- O usuário deverá estar cadastrado.

Como Acessar

Acessar o portal > Realizar Login > Avisos > Ações > [UC016] Excluir aviso

História de Usuário

COMO monitor, professor e administrador

QUERO excluir avisos cadastrados

PARA não terem mais acesso a eles.

Cenários

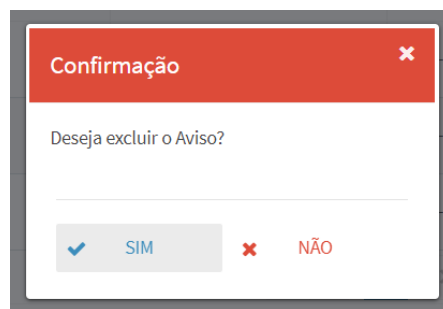
Código	Descrição	Tipo	Cenário	Referência
CN-01	Excluir aviso	Excluir	DADO QUE é usuário específico do sistema, QUANDO clica no ícone de excluir de um aviso, E clica em “SIM”; ENTÃO visualiza a mensagem de: “Monitoria Excluída com sucesso”.	

Critérios de Aceitação

Código	Descrição	Aceito?

Protótipos

P1 – Excluir Aviso



Descrição dos Campos

Nome	Descrição	Tipo	Tipo de Dado (tamanho)	Editável	Obrigatório	Regras
Confirmação	Input text	Text	Texto	Não	Sim	Apenas uma mensagem para confirmação da exclusão

Descrição dos Comandos

Nome	Ação	Tipo	Restrições
SIM	O aviso será excluído do sistema.	Botão	O botão será clicável
Não	Sairá da página de exclusão.	Botão	O botão será clicável

Regras de Apresentação

Descrição
<ul style="list-style-type: none">N/A

[UC017] Listar avisos

Usuários/Perfis

Aluno, monitor, professor e administrador.

Pré-Condições

- O usuário deverá estar logado.

Como Acessar

Acessar o portal > Realizar Login > Avisos > [UC017] Listar avisos

História de Usuário

COMO aluno, monitor, professor e administrador

QUERO listar avisos

PARA visualizar todos avisos cadastrados.

Cenários







Código	Descrição	Tipo	Cenário	Referência
CN-01	Listar avisos	Listagem	DADO QUE é usuário do sistema, QUANDO clica em “Avisos”, ENTÃO visualiza todos avisos cadastrados.	
CN-02	Sem aviso cadastrado	Erro	DADO QUE é usuário do sistema, QUANDO clica em “Avisos”, ENTÃO visualiza a mensagem: “Nenhuma Aviso cadastrado”.	

Critérios de Aceitação

Código	Descrição	Aceito?

Protótipos

P1 – Listar Avisos

Assunto	Descrição	Monitoria	Data	Autor	Opções
wert	qwerty	TCC II - 2018/02	2018-12-03	Vinicius Mateus Martins	 
j	j	TCC II - 2018/02	2018-12-05	Fábio	 
hg	h	TCC II - 2018/02	2018-12-06	Jackeline	 

Descrição dos Campos

Nome	Descrição	Tipo	Tipo de Dado (tamanho)	Editável	Obrigatório	Regras
Assunto	Campo mostra o assunto do aviso	Text	Texto	Não	Sim	Somente visualização dos dados cadastrados.
Descrição	Campo mostra descrição do aviso	Text	Texto	Não	Sim	Somente visualização dos dados cadastrados.
Monitoria	Campo mostra qual a monitoria do aviso	Text	Texto	Não	Sim	Somente visualização dos dados cadastrados.
Data	Campo mostra a data do aviso	Text	Texto	Não	Sim	Somente visualização dos dados cadastrados.
Autor	Campo mostra o autor do aviso	Text	Texto	Não	Sim	Somente visualização dos dados cadastrados.
Opções	Campo mostra as ações permitidas	Selection	seleção	Não	Sim	N/A

Descrição dos Comandos

Nome	Ação	Tipo	Restrições

Regras de Apresentação

Descrição
<ul style="list-style-type: none"> N/A

Fórum de Discussão

[FORUM DE DISCUSSÃO] Ativar Fórum

COMO administrador do sistema

QUERO ativar Fórum

PARA desbloquear o acesso ao Fórum

CENARIO 1: Ativar fórum

DADO que estou logado no sistema

E o Fórum está bloqueado

QUANDO acesso as configurações do Fórum

E clico em Ativar Fórum

E clico em salvar

ENTÃO o Fórum é desbloqueado

[FORUM DE DISCUSSÃO] Desativar Fórum

COMO administrador do sistema

QUERO desativar Fórum

PARA bloquear todas as atividades deste

CENARIO 1: Desativar Fórum

DADO que estou logado no sistema

E o fórum está desbloqueado

QUANDO acesso as configurações do Fórum

E clico em Desativar Fórum

E clico em salvar

ENTÃO o Fórum é bloqueado

[FORUM DE DISCUSSÃO] Editar Fórum

COMO Administrador do sistema

QUERO editar Fórum

PARA alterar as configurações do Fórum

CENARIO 1: Editar Fórum

DADO que estou logado no sistema

QUANDO acesso as configurações do Fórum

E efetuo as edições necessárias

E clico em salvar

ENTÃO as edições são salvas

CENARIO 2: Erro ao Editar Fórum

DADO que estou logado no sistema

QUANDO acesso as configurações do Fórum

E efetuo as edições necessárias

E deixo campos obrigatórios em branco

E clico em salvar

ENTÃO exibe uma mensagem informando o erro

[FORUM DE DISCUSSÃO] Cadastrar Pergunta

COMO qualquer usuário do sistema

QUERO cadastrar pergunta

PARA gerar discussões sobre a pergunta e sanar dúvidas

CENARIO 1: Cadastrar pergunta

DADO que estou logado no sistema

E o Fórum está ativo

QUANDO acesso a aba do Fórum

E clico em Cadastrar Pergunta

E escrevo a pergunta desejada

E clico em publicar

ENTÃO a pergunta é publicada

CENARIO 2: Erro ao cadastrar pergunta

DADO que estou logado no sistema

E o Fórum está ativo

QUANDO acesso a aba do Fórum

E clico em Cadastrar Pergunta

E deixo campos obrigatórios em branco

E clico em publicar

ENTÃO exibe uma mensagem informando o erro

[FORUM DE DISCUSSÃO] Editar Pergunta

COMO qualquer usuário do sistema

QUERO editar pergunta

PARA alterar a pergunta já publicada por mim

CENARIO 1: Editar pergunta

DADO que estou logado no sistema

E existem perguntas publicadas

QUANDO acesso a aba do Fórum

E clico na pergunta que desejo editar

E que foi publicada por mim

E clico em editar

E faço as edições desejadas

E clico em salvar

ENTÃO a pergunta é salva

CENARIO 2: Erro ao editar pergunta

DADO que estou logado no sistema

E existem perguntas publicadas

QUANDO acesso a aba do Fórum

E clico na pergunta que desejo editar

E que foi publicada por mim

E clico em editar

E faço as edições desejadas

E deixo campos obrigatórios em branco

E clico em salvar
ENTÃO exibe uma mensagem informando o erro

[FORUM DE DISCUSSÃO] Buscar Pergunta

COMO qualquer usuário do sistema
QUERO buscar pergunta
PARA localizar a pergunta desejada

CENARIO 1: Buscar pergunta

DADO que estou logado no sistema
E existem perguntas publicadas
QUANDO acesso ao Fórum
E digito na aba de busca
E clico em buscar
E forem encontrados os resultados
ENTÃO é retornado todos os resultados da busca

CENARIO 2: Erro ao buscar pergunta

DADO que estou logado no sistema
E existem perguntas publicadas
QUANDO acesso ao Fórum
E digito na aba de busca
E clico em buscar
E não forem encontrados os resultados
ENTÃO é retornado uma mensagem que nenhum dado foi encontrado

[FORUM DE DISCUSSÃO] Excluir Pergunta

COMO qualquer usuário do sistema
QUERO excluir pergunta
PARA apagar perguntas já publicadas por mim

CENARIO 1: Excluir pergunta

DADO que estou logado no sistema
E existem perguntas publicadas
QUANDO acesso a aba do Fórum
E clico na pergunta que desejo excluir
E que foi publicada por mim
E clico em excluir
ENTÃO a pergunta é excluída

[FORUM DE DISCUSSÃO] Listar Pergunta

COMO qualquer usuário do sistema
QUERO listar perguntas
PARA mostrar todas as perguntas publicadas

CENARIO 1: Listar perguntas

DADO que estou logado no sistema
E existem perguntas publicadas
QUANDO clico na aba do Fórum

ENTÃO é listado todas as perguntas publicadas

CENARIO 2: Erro ao listar perguntas

DADO que estou logado no sistema

E não exista perguntas publicadas

QUANDO clico na aba do Fórum

ENTÃO exibe uma mensagem de que nenhuma pergunta foi publicada

[FORUM DE DISCUSSÃO] Cadastrar Resposta

COMO qualquer usuário do sistema

QUERO cadastrar resposta

PARA responder uma pergunta publicada

CENARIO 1: Cadastrar resposta

DADO que estou logado no sistema

E existem perguntas publicadas

QUANDO acesso a aba do Fórum

E clico na pergunta desejada

E escrevo uma resposta

E clico em Responder

ENTÃO a resposta é publicada

CENARIO 2: Erro ao cadastrar resposta

DADO que estou logado no sistema

E existem perguntas publicadas

QUANDO acesso a aba do Fórum

E clico na pergunta desejada

E deixo campos obrigatórios em branco

E clico em Responder

ENTÃO exibe uma mensagem informando o erro

[FORUM DE DISCUSSÃO] Editar Resposta

COMO qualquer usuário do sistema

QUERO editar resposta

PARA alterar a resposta já publicada por mim

CENARIO 1: Editar resposta

DADO que estou logado no sistema

E existem perguntas publicadas

E existem respostas publicadas

QUANDO acesso a aba do Fórum

E clico na pergunta desejada

E clico na resposta que desejo editar

E que foi publicada por mim

E clico em editar

E faço as edições desejadas

E clico em salvar

ENTÃO a resposta é salva

CENÁRIO 2: Erro ao editar resposta

DADO que estou logado no sistema
E existem perguntas publicadas
E existem respostas publicadas
QUANDO acesso a aba do Fórum
E clico na pergunta desejada
E clico na resposta que desejo editar
E que foi publicada por mim
E clico em editar
E faço as edições desejadas
E deixo campos obrigatórios em branco
E clico em Salvar
ENTÃO exibe uma mensagem informando o erro

[FORUM DE DISCUSSÃO] Excluir Resposta

COMO qualquer usuário do sistema
QUERO excluir resposta
PARA apagar uma resposta já publicada por mim

CENÁRIO 1: Excluir Resposta

DADO que estou logado no sistema
E existem perguntas publicadas
E existem respostas publicadas
QUANDO acesso a aba do Fórum
E clico na pergunta desejada
E clico na resposta que desejo excluir
E que foi publicada por mim
E clico em excluir
ENTÃO a resposta é excluída permanentemente

Manter Publicação

[MANTER PUBLICAÇÃO] Enviar Publicação

COMO monitor, professor ou administrador do sistema
QUERO enviar uma publicação
PARA disponibilizar a todos os usuários

CENÁRIO 1: Enviar publicação

DADO que estou logado no sistema
E entro em uma disciplina
E tem usuários cadastrados
QUANDO clico em enviar publicação
E preencho todos os campos
E seleciono os usuários
E clico em enviar
ENTÃO a publicação é enviada

CENARIO 2: Erro ao enviar publicação

DADO que estou logado no sistema

E entro em uma disciplina

E tem usuários cadastrados

QUANDO clico em enviar publicação

E não preencho todos os campos

E seleciono os usuários

E clico em enviar

ENTÃO é emitido uma mensagem informando o erro

CENARIO 3: Erro ao enviar publicação

DADO que estou logado no sistema

E entro em uma disciplina

E tem usuários cadastrados

QUANDO clico em enviar publicação

E preencho todos os campos

MAS não seleciono os usuários

E clico em enviar

ENTÃO é emitido uma mensagem informando o erro

[MANTER PUBLICAÇÃO] Editar Publicação

COMO monitor, professor ou administrador do sistema

QUERO editar uma publicação minha

PARA que as informações sejam atualizadas

CENARIO 1: Editar publicação

DADO que estou logado no sistema

E existem publicações cadastradas

QUANDO clico em publicações

E abre as publicações

E seleciono uma publicação feita por mim

E clico em editar

E altero a publicação

E clico em Salvar

ENTÃO a publicação é salva

CENARIO 2: Erro ao editar publicação

DADO que estou logado no sistema

E existem publicações cadastradas

QUANDO clico em publicações

E abre as publicações

E seleciono uma publicação feita por mim

E clico em editar

E não preencho todos os campos

E clico em Salvar

ENTÃO exibe uma mensagem informando o erro

[MANTER PUBLICAÇÃO] Excluir Publicação

COMO monitor, professor ou administrador do sistema

QUERO excluir uma publicação minha

PARA que as informações desta publicação não seja mais ser visualizada

CENARIO 1: Excluir publicação

DADO que estou logado no sistema
E existem publicações cadastradas
QUANDO clico em publicações
E abre todas as publicações
E clico em minha publicação
E clico em Excluir
ENTÃO a publicação é excluída

[MANTER PUBLICAÇÃO] Listar Publicação

COMO aluno, monitor, professor ou administrador do sistema
QUERO listar as publicações
PARA visualizar todas as publicações existentes

CENARIO 1: Listar publicação

DADO que estou logado no sistema
E existem publicações cadastradas
QUANDO clico em publicações
E abre todas as publicações
ENTÃO posso visualizar uma lista com todas as publicações existentes

CENARIO 2: Sem publicações

DADO que estou logado no sistema
E não há publicações cadastradas
QUANDO clico em publicações
ENTÃO é emitido uma mensagem de aviso sem publicações cadastradas

Manter Chat

[MANTER CHAT] Iniciar Chat

COMO aluno, monitor, professor ou administrador do sistema
QUERO iniciar uma conversa no chat
PARA tirar minhas dúvidas

CENARIO 1: Iniciar chat

DADO que estou logado no sistema
E há usuários cadastrados
QUANDO clico na aba chat
E escolho um usuário para conversar
E clico no usuário
ENTÃO é aberta uma caixa privada para o envio de mensagens

[MANTER CHAT] Deletar Chat

COMO aluno, monitor, professor ou administrador do sistema
QUERO deletar uma conversa do chat
PARA limpar minha caixa de mensagens

CENARIO 1: Deletar chat

DADO que estou logado no sistema

E há usuários cadastrados

QUANDO abro a aba chat

E existir conversas

E seleciono a conversar a ser excluída

E clico em Excluir

ENTÃO todas as mensagens da conversa são apagadas

Candidatar a Monitoria

[CANDIDATAR A MONITORIA] Candidatar a Monitor

COMO aluno do sistema

QUERO me candidatar a monitor

PARA auxiliar o professor na disciplina ministrada

CENARIO 1: Candidatar a monitor

DADO que estou logado no sistema

E há disciplinas cadastradas

QUANDO clico na aba disciplinas

E abre uma lista com todas as disciplinas

E seleciono uma disciplina

E abre os requisitos da disciplina

E clico em Candidatar a Monitoria

ENTÃO é enviado os dados do aluno para o professor aprovar

[CANDIDATAR A MONITORIA] Cancelar Candidatura

COMO aluno do sistema

QUERO cancelar meu pedido de candidatura a monitoria

PARA que o professor não precise avaliar

CENARIO 1: Cancelar candidatar

DADO que estou logado no sistema

E há disciplinas cadastradas

QUANDO clico na aba disciplinas

E abre uma lista com todas as disciplinas

E seleciono uma disciplina que sou candidato

E abre os requisitos da disciplina

E clico em Cancelar Candidatura

ENTÃO a candidatura é cancelada

Manter Chamada

[MANTER CHAMADA] Fazer Chamada

COMO monitor, professor ou administrador do sistema

QUERO fazer chamada
PARA armazenar a frequência dos alunos

CENARIO 1: Fazer chamada

DADO que estou logado no sistema
QUANDO acesso a aba de disciplina
E seleciono a disciplina desejada
E clico em Fazer Chamada
E preencho os campos necessários
E clico em salvar
ENTÃO a chamada é salva no sistema

CENARIO 2: Erro ao fazer chamada

DADO que estou logado no sistema
QUANDO acesso a aba de disciplina
E seleciono a disciplina desejada
E clico em Fazer Chamada
E preencho os campos necessários
E deixo algum campo obrigatório em branco
E clico em salvar
ENTÃO exibe uma mensagem informando o erro

[MANTER CHAMADA] Editar Chamada

COMO monitor, professor ou administrador do sistema
QUERO editar chamada
PARA alterar a frequência dos alunos de uma chamada já feita

CENARIO 1: Editar chamada

DADO que estou logado no sistema
QUANDO acesso a aba de disciplina
E a lista de frequência estiver preenchida
E esteja dentro de um período de 24 horas desde sua publicação
E seleciono a disciplina desejada
E clico em Fazer Chamada
E altero os campos necessários
E clico em salvar
ENTÃO a chamada é atualizada no sistema

CENARIO 2: Erro ao editar chamada

DADO que estou logado no sistema
QUANDO acesso a aba de disciplina
E a lista de frequência estiver preenchida
E esteja dentro de um período de 24 horas desde sua publicação
E seleciono a disciplina desejada
E clico em Fazer Chamada
E altero os campos necessários
E deixo algum campo obrigatório em branco
E clico em salvar
ENTÃO exibe uma mensagem informando o erro

[MANTER CHAMADA] Exportar Lista de Frequência

COMO professor ou administrador do sistema

QUERO Exportar Lista de Frequência

PARA obter o relatório de frequência dos alunos

CENÁRIO 1: Exportar lista de frequência

DADO que estou logado no sistema

QUANDO acesso a aba de disciplina

E seleciono a disciplina desejada

E houver frequências cadastradas

E clico em Exportar Lista de Frequência

ENTÃO é emitido um relatório da frequência dos alunos

CENÁRIO 2: Sem frequência cadastradas

DADO que estou logado no sistema

QUANDO acesso a aba de disciplina

E seleciono a disciplina desejada

E não houver frequências cadastradas

ENTÃO exibe uma mensagem de aviso que não há lista de frequência para esta disciplina

Aprovar Monitor

[APROVAR MONITOR] Aprovar Monitor

COMO professor do sistema

QUERO aprovar o pedido do candidato a monitor

PARA que minha disciplina possua um monitor disponível aos alunos

CENÁRIO 1: Aprovar monitor

DADO que estou logado no sistema

QUANDO recebo o cadastro do candidato a monitor

E clico no cadastro

E verifico se está tudo em ordem

E se atende aos requisitos da vaga

E ainda tiver vaga disponível

E clico em Aprovar

ENTÃO o candidato está aprovado como monitor

CENÁRIO 2: Sem vagas para monitor

DADO que estou logado no sistema

QUANDO recebo o cadastro do candidato a monitor

E clico no cadastro

E verifico se está tudo em ordem

E se atende aos requisitos da vaga

MAS não há vaga disponível

E clico em Aprovar

ENTÃO é emitido uma mensagem informando o erro

[APROVAR MONITOR] Reprovar Monitor

COMO professor do sistema

QUERO reprovar o pedido de candidatura a monitor

PARA que o candidato não seja monitor da minha disciplina

CENARIO 1: Reprovar monitor

DADO que estou logado no sistema

QUANDO recebo o cadastro do candidato a monitor

E clio no cadastro

E não atende todos os requisitos

E clio em Reprovar

ENTÃO o candidato é reprovado como monitor

Emitir Ficha de Avaliação

[EMITIR FICHA DE AVALIAÇÃO] Emitir Ficha de Avaliação

COMO professor do sistema

QUERO emitir ficha de avaliação de monitoria

PARA avaliar o monitor e os alunos que participam das monitorias

CENARIO 1: Emitir ficha

DADO que estou logado no sistema

QUANDO clio em Emitir ficha de Avaliação

E abre a ficha de avaliação

E insiro os dados necessários nos campos

E clio em Emitir

ENTÃO é emitido as fichas de avaliações

CENARIO 2: Erro ao emitir ficha

DADO que estou logado no sistema

QUANDO clio em Emitir ficha de Avaliação

E é aberta a ficha de avaliação

E insiro os dados necessários nos campos

E deixo algum campo obrigatório em branco

E clio em Emitir

ENTÃO exibe uma mensagem informando o erro

Aprovar Emissão de Certificado

[APROVAR EMISSÃO DE CERTIFICADO] Aprovar Certificado

COMO administrador do sistema

QUERO aprovar a emissão de certificados dos monitores

PARA que eles possam ser validados e impressos

CENARIO 1: Aprovar emissão de certificado

DADO que estou logado no sistema

QUANDO clio na aba "Certificados "

E é aberta uma página com as disciplinas que possuem certificados pendentes
E identifico que o monitor cumpriu com a monitoria
E clico em Aprovar
ENTÃO o certificado fica disponível para ser emitido e impresso pelo monitor

[APROVAR EMISSÃO DE CERTIFICADO] Emitir Certificado

COMO monitor do sistema
QUERO emitir certificados aprovados
PARA serem impressos

CENARIO 1: Emitir certificado

DADO que estou logado no sistema
QUANDO clico na aba Certificados
E abre uma página com os certificados das minhas monitorias
E identifico que existem certificados aprovados
E clico em Emitir
ENTÃO o certificado é baixado

CENARIO 2: Sem certificados

DADO que estou logado no sistema
QUANDO clico na aba Certificados
E abre uma página com os certificados das minhas monitorias
E não existem certificados aprovados
ENTÃO é exibido uma mensagem dizendo que não há certificados

Manter Edital

[MANTER EDITAL] Cadastra Edital

COMO Administrador do sistema
QUERO cadastra o edital
PARA que fique disponível para o acesso

CENARIO 1: Cadastrar edital

DADO que já estou logado no sistema
QUANDO clico na aba Edital
E não houver nenhum edital cadastrado
E clico em cadastrar edital
E preencho os campos necessários
E clico em Salvar
ENTÃO o edital é cadastrado

CENARIO 2: Edital existente

DADO que já estou logado no sistema
QUANDO clico na aba Edital
E há um edital cadastrado
E clico em Cadastrar
ENTÃO é exibido uma mensagem de que já existe um edital

CENARIO 3: Erro ao emitir edital

DADO que já estou logado no sistema

QUANDO clico na aba Edital
E não houver nenhum edital cadastrado
E clico em Cadastrar Edital
E preencho os campos necessários
E deixo algum campo obrigatório em branco
E clico em Salvar
ENTÃO exibe uma mensagem informando o erro
[MANTER EDITAL] Editar Edital
COMO Administrador do sistema
QUERO editar Edital
PARA alterar informações de editais já emitidos

CENARIO 1: Editar edital

DADO que já estou logado no sistema
QUANDO clico na aba Edital
E houver um edital cadastrado
E clico em editar edital
E preencho os campos necessários
E clico em Salvar
ENTÃO as informações são alteradas

CENARIO 2: Sem edital

DADO que já estou logado no sistema
QUANDO clico na aba Edital
E não houver um edital cadastrado
E clico em editar edital
ENTÃO é exibido a mensagem de que não existe edital cadastrado

CENARIO 3: Erro ao editar edital

DADO que já estou logado no sistema
QUANDO clico na aba Edital
E houver um edital cadastrado
E clico em editar edital
E preencho os campos necessários
E deixo algum campo obrigatório em branco
E clico em Salvar
ENTÃO exibe uma mensagem informando o erro

[MANTER EDITAL] Excluir Edital

COMO Administrador do sistema
QUERO excluir edital
PARA deletar um edital cadastrado

CENARIO 1: Excluir edital

DADO que estou logado no sistema
QUANDO clico na aba Edital
E houver um edital cadastrado
E clico em excluir edital
ENTÃO o edital é excluído permanentemente

CENARIO 2: Sem edital

DADO que estou logado no sistema

QUANDO clico na aba Edital

E não houver um edital cadastrado

E clico em excluir edital

ENTÃO é exibido uma mensagem de que não há edital para excluir

APÊNDICE B – DOCUMENTO DE REQUISITOS NÃO-FUNCIONAIS ²

Identificador	RNF001	Categoria	Desempenho
Nome	Tempo limite de resposta do sistema		
Data de Criação	17/09/2018	Autor	Wanderson Inácio
Data da última alteração	N/A	Autor	N/A
Versão	1.0	Prioridade	Média
Descrição	<p>O Sistema deve atender a um determinado tempo de resposta para satisfazer ao usuário. Em função desta realidade, o sistema deverá prover um processamento rápido que possibilite o tempo de execução de determinadas páginas.</p> <p>Considerando a satisfação do usuário, o sistema deve processar e dar alguma resposta em menos de 3 segundos.</p>		

Identificador	RNF002	Categoria	Segurança
Nome	Autenticação de usuário		
Data de Criação	17/09/2018	Autor	Wanderson Inácio
Data da última alteração	N/A	Autor	N/A
Versão	1.0	Prioridade	Alta
Descrição	<p>Para a autenticação de cada usuário para acessar o nível de aplicação, é necessário que tenha um cadastro válido na base de dados do sistema.</p> <p>Deverá haver uma política de segurança que assegure a autenticidade de cada um dos usuários.</p>		

Identificador	RNF003	Categoria	Usabilidade
Nome	Design simples nas interfaces gráficas		
Data de Criação	17/09/2018	Autor	Wanderson Inácio
Data da última alteração	N/A	Autor	N/A
Versão	1.0	Prioridade	Média
Descrição	<p>O sistema de monitoria será construído para rodar em ambiente Web. Deverá ter um design responsivo (https://www.secnet.com.br/blog/o-que-e-layout-responsivo).</p> <p>A interface deverá se comportar adequadamente independente do front-end a ser utilizado para acesso.</p>		

Identificador	RNF004	Categoria	Compatibilidade
----------------------	--------	------------------	-----------------

² Por estar utilizando uma metodologia iterativa e incremental, esta é a 1ª versão.

Nome	Compatibilidade com Navegador Web		
Data de Criação	17/09/2018	Autor	Wanderson Inácio
Data da última alteração	N/A	Autor	N/A
Versão	1.0	Prioridade	Média
Descrição	O sistema por se tratar de uma página Web, deverá ser compatível com os principais navegadores.		

Identificador	RNF005	Categoria	Padrão
Nome	Divisão arquitetural no padrão MVC (<i>Model View Controller</i>)		
Data de Criação	17/09/2018	Autor	Wanderson Inácio
Data da última alteração	N/A	Autor	N/A
Versão	1.0	Prioridade	Alta
Descrição	<p>O sistema será desenvolvido com a arquitetura separada em camadas, onde cada uma das camadas será responsável pelos algoritmos associados à sua camada.</p> <p>A arquitetura utilizada será o MVC, no qual o:</p> <ul style="list-style-type: none"> • <i>Model</i> – responsável pela manipulação de dados como leitura, escrita de dados e validações; • <i>View</i> – responsável pela interação com o usuário, faz apenas exibição dos dados sendo em HTML ou XML; • <i>Controller</i> – responsável por receber as requisições do usuário, exerce o controle de qual <i>Model</i> deverá ser aplicado e de qual <i>View</i> será mostrado ao usuário. 		

Academicci

Documento de Arquitetura de Software

Versão 0.005

³ Por estar utilizando uma metodologia iterativa e incremental, esta é a 4ª versão.

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 Monitória • 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.

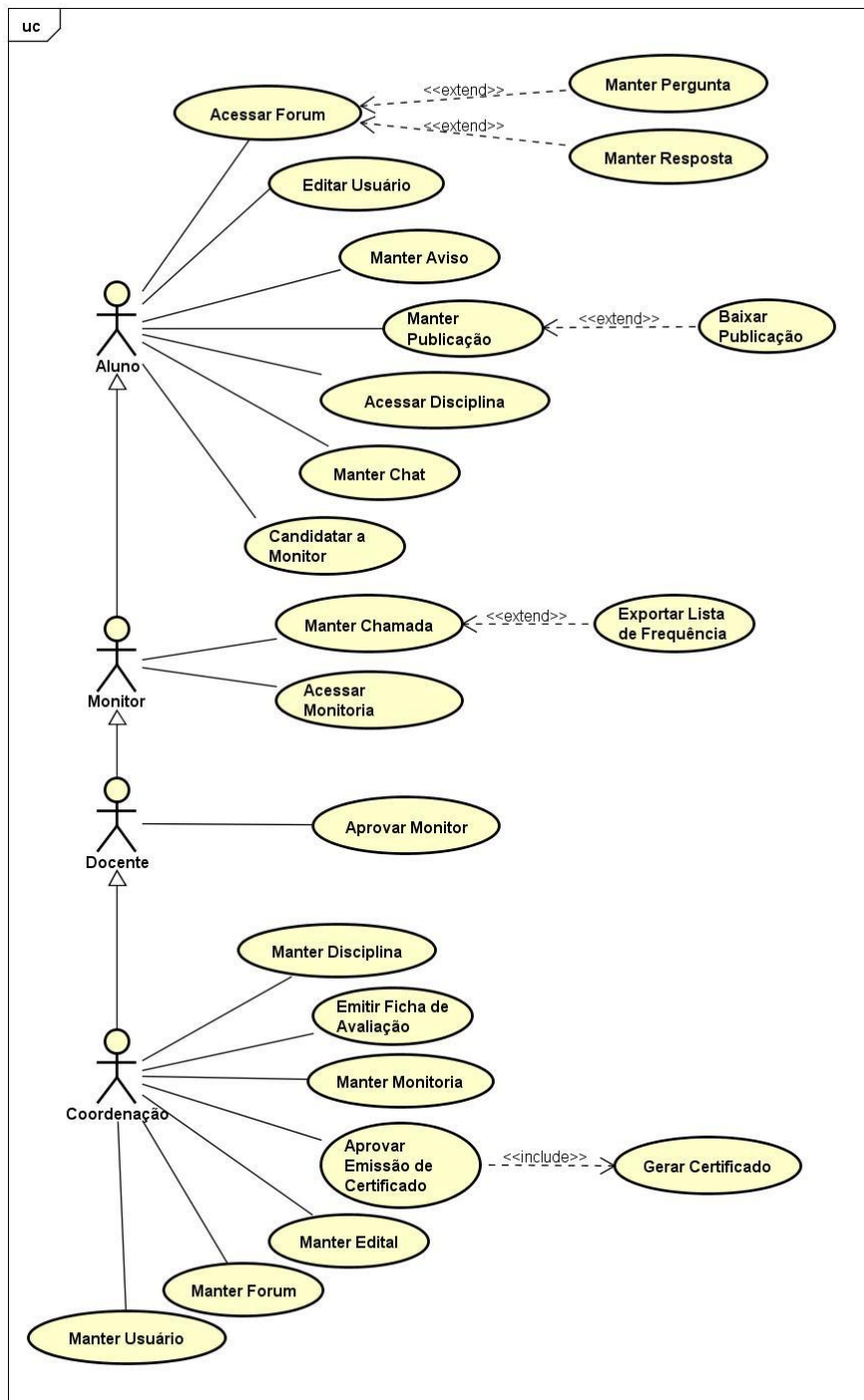
8. Metas e Restrições da Arquitetura

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. Conterá 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

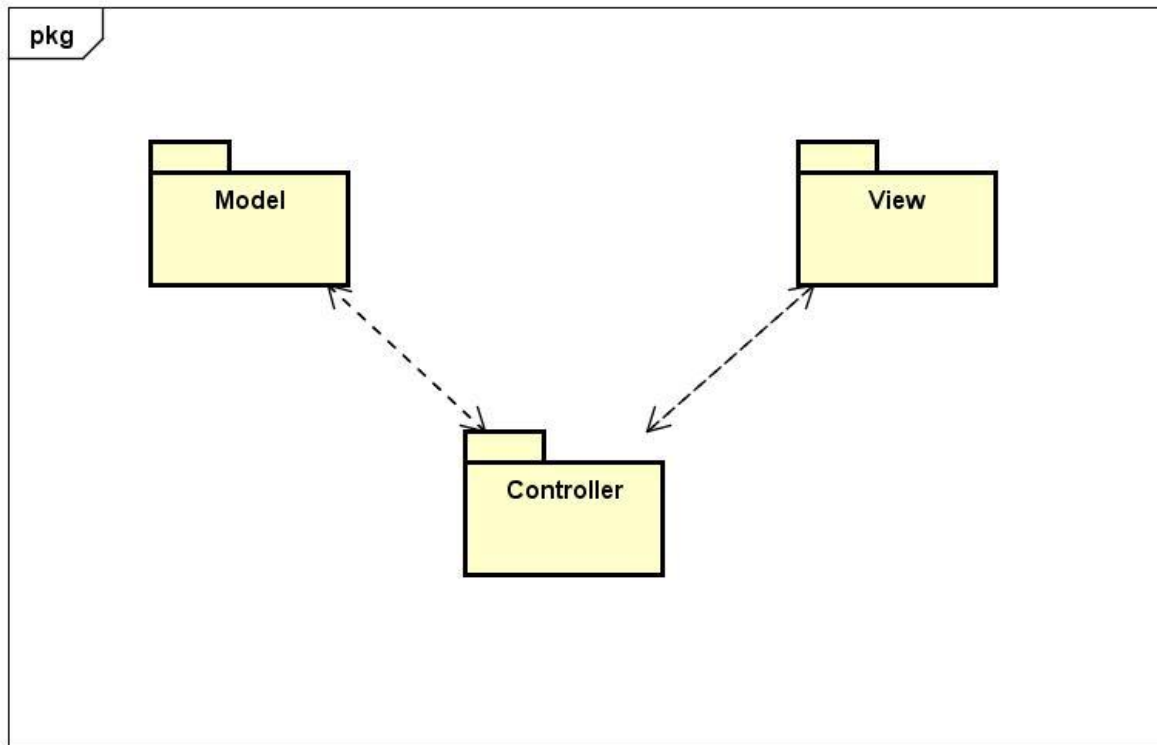
Caso de Uso	Descrição
Acessar Forum	Visualizar as publicações do Forum
Manter Pergunta	Criar, alterar ou excluir perguntas no Forum
Manter Resposta	Criar, alterar ou excluir respostas no Forum
Editar Usuário	Alterar as informações do usuário
Manter Aviso	Criar, alterar, excluir, listar, expirar um aviso
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

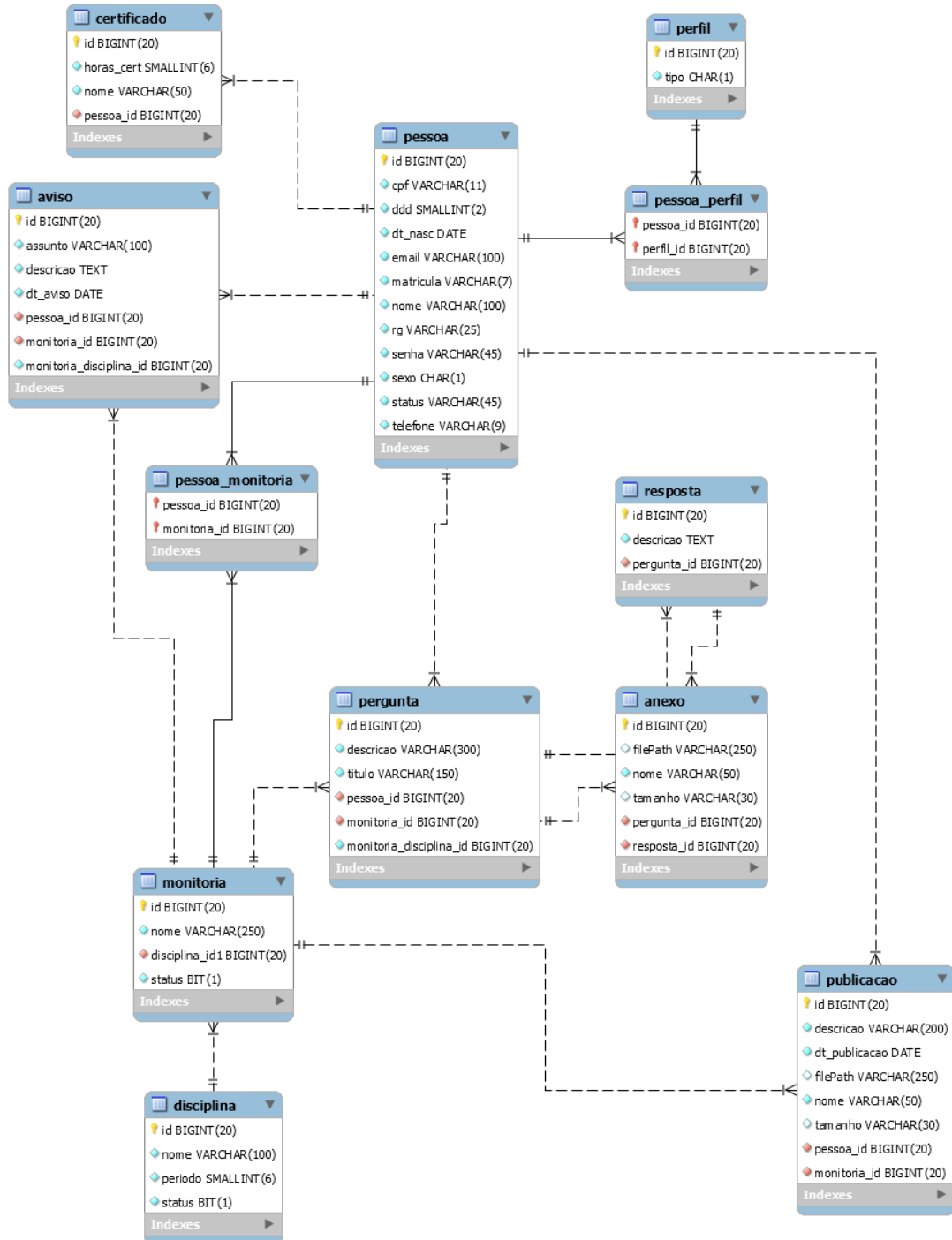
A camada controller é responsável por comunicar as requisições do usuário ao sistema.

11.2.3. Model

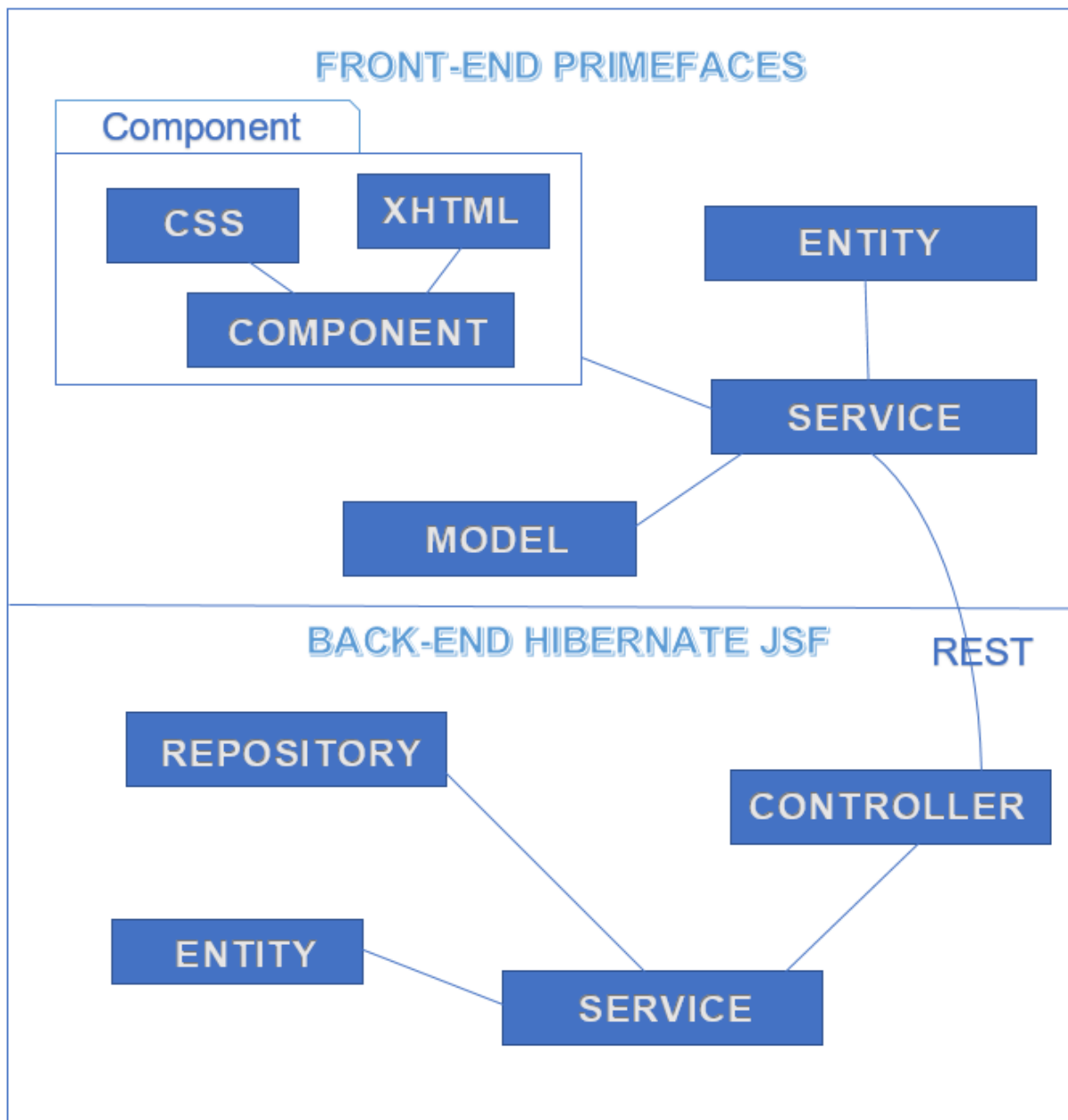
A camada model é responsável pelo armazenamento das classes no banco de dados.

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