

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO
GRANDE DO NORTE
CAMPUS NATAL – ZONA NORTE
CURSO TÉCNICO INTEGRADO EM INFORMÁTICA

ALISON HEDIGLIRANES DA SILVA
LUCAS PEDRO MEDEIROS E BARBOSA
TIAGO CAIO DE OLIVEIRA LISBOA

ACHA: ASSISTENTE DE CARGA HORÁRIA ACADÊMICA

NATAL/RN
2017

ALISON HEDIGLIRANES DA SILVA
LUCAS PEDRO MEDEIROS E BARBOSA
TIAGO CAIO DE OLIVEIRA LISBOA

ACHA: ASSISTENTE DE CARGA HORÁRIA ACADÊMICA

Trabalho de conclusão de curso apresentado a Coordenação do Curso Técnico Integrado de Informática do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte - Campus Natal-Zona Norte, como forma de cumprir os requisitos para a obtenção do certificado de conclusão de curso.

Orientador(ar): Prof. MSc. Edmilson Barbalho Campos Neto
Coorientador(ar): Prof^a. MSc. Alba Sandrya Bezerra Lopes

ALISON HEDIGLIRANES DA SILVA
LUCAS PEDRO MEDEIROS E BARBOSA
TIAGO CAIO DE OLIVEIRA LISBOA

ACHA: ASSISTENTE DE CARGA HORÁRIA ACADÊMICA

Trabalho de conclusão de curso apresentado a Coordenação do Curso Técnico Integrado de Informática do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte - Campus Natal-Zona Norte, como forma de cumprir os requisitos para a obtenção do certificado de conclusão de curso.

Aprovado em: 10 / 102 / 2017.

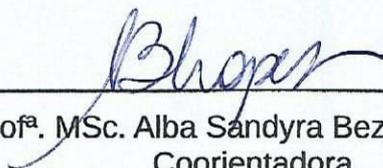


Prof. MSc. Edmilson Barbalho Campos Neto
Coordenador do curso

BANCA EXAMINADORA



Prof. MSc. Edmilson Barbalho Campos Neto
Orientador



Prof. MSc. Alba Sandyra Bezerra Lopes
Coorientadora

Prof. PhD. Diego Silveira Costa Nascimento
Avaliador

Natal/RN
2017

*“Nem tudo que é ouro fulgura,
Nem todo o vagante é vadio;
O velho que é forte perdura,
Raiz funda não sofre o frio.”*

(J.R.R. Tolkien)

RESUMO

Desde da criação do IFRN Campus Natal Zona Norte o processo de criação de matrizes e distribuição de carga horária para os professores, quando realizado, é feito de forma manual, o que acarreta uma série de problemas como a demora para conclusão das tarefas e uma margem alta para erros, e com o passar do tempo se fez necessário automatizar esses processos já que cada vez mais turmas e professores precisam ser gerenciados e pensando em resolver esse problemas desenvolvemos o ACHA (assistente de carga horária acadêmica). O sistema permite a organização e manipulação de diversos dados importantes para o funcionamento da instituição acadêmica, como informações sobre: professores, disciplinas, atividades acadêmicas, cursos, turmas e matrizes. O conjunto desses possibilita ao usuário, através do *software*, projetar o comportamento da academia em relação a oferta de turmas com matrizes variadas, definidas previamente, cada qual com suas necessidades de carga horária e, conseqüentemente professores. Para isso, utiliza-se um esquema de criação de turmas simuladas de forma simples e intuitiva - através de uma tabela com elementos arrastáveis, que são as turmas - que possibilitam a geração de dados acerca do quadro de professores nos períodos escolares abordados, permitindo que a administração do campus atendam as necessidades da comunidade dentro das possibilidades da instituição. Para atingir esse fim, foi utilizada a linguagem PHP para manipulação de dados, um banco de dados em MySQL, e uma aplicação web criada através do *framework* angularjs com a biblioteca de css Bootstrap, que facilita a criação de interfaces amigáveis e responsivas. Ao longo do processo de desenvolvimento foram aplicadas algumas técnicas de engenharia de *software*, principalmente nas etapas de projeto, visando, além da construção do objeto tratado neste documento, o aprendizado dos discentes incubidos desta tarefa.

Palavras-chave: Carga Horária. IFRN. Sistema de apoio.

SUMÁRIO

1	INTRODUÇÃO.....	6
1.1	JUSTIFICATIVA.....	9
1.2	OBJETIVOS.....	9
1.3	RESULTADOS ESPERADOS.....	10
1.4	METODOLOGIA.....	10
1.5	ORGANIZAÇÃO DO DOCUMENTO.....	11
2	FUNDAMENTAÇÃO TEÓRICA.....	13
2.1	ENGENHARIA DE SOFTWARE.....	13
2.1.1	Planejamento e gerenciamento.....	14
2.1.2	Processo unificado.....	15
2.1.3	UML.....	15
2.1.4	MVC.....	15
2.2	SISTEMAS DE APOIO À DECISÃO.....	16
2.3	FRAMEWORK.....	17
2.4	SUMÁRIO.....	17
3	SOLUÇÃO PROPOSTA.....	18
3.1	LEVANTAMENTO DE REQUISITOS.....	18
3.2	MODELAGEM DO SISTEMA.....	22
3.2.1	Casos de uso.....	22
3.2.2	Arquitetura do sistema.....	26
3.2.3	Protótipos de baixa fidelidade.....	28
3.2.4	Protótipos de alta fidelidade.....	30
3.3	MODELAGEM DE BANCO DE DADOS.....	31
3.4	LINGUAGENS E FERRAMENTAS UTILIZADAS.....	32
3.5	ORGANIZAÇÃO DO GRUPO.....	34

3.6 SUMÁRIO.....	35
4 RESULTADOS OBTIDOS.....	36
4.1 FUNCIONALIDADES DO SISTEMA.....	36
4.1.1 Planejamentos e Viabilidade.....	36
4.1.2 Matrizes.....	39
4.1.3 Professores.....	39
4.1.4 Grupos, áreas e disciplinas.....	40
4.2 SUMÁRIO.....	42
5 CONSIDERAÇÕES FINAIS.....	43
5.1 ATIVIDADES FUTURAS.....	43
REFERÊNCIAS.....	44
ANEXO I - GLOSSÁRIO.....	47

1 INTRODUÇÃO

Quando se pensa em organizações de grande porte, como empresas e instituições diversas, é comum imaginar a dificuldade de gerenciá-las. E isso é bem verdade. O segredo do sucesso desses grupos está, na maioria das vezes, em um corpo gestor grande e capaz.

Entretanto, é preciso perceber que a administração não se aplica apenas ao ambiente empresarial, uma vez que “administração é um processo distinto, que consiste no planejamento, organização, atuação e controle, para determinar e alcançar os objetivos da organização pelo uso de pessoas e recursos.” (TERRY, 1953).

Isso pode ser observado no ambiente escolar, por exemplo. Nele temos as diversas turmas, as quais possuem uma certa demanda por professores. Nesse panorama, fica a cargo da administração da instituição de ensino distribuir o quadro de professores pelas turmas, atentando a necessidade das turmas pelas disciplinas e ao acúmulo de carga horária pelos professores. Esse quadro fica ainda mais complicado quando se leva em conta escolas onde os professores podem orientar projetos de pesquisa, acumulando carga horária fora da sala de aula.

Expondo isso, pode-se atentar para a necessidade de uma boa gestão do corpo docente. E, conforme explicitado anteriormente, uma das etapas da administração é o planejamento que, segunda a definição de Dalmás (2008, p. 24) “[...] é uma tentativa de atingir e ordenar decisões que deverão ser tomadas visando a atingir algum conjunto de objetivos específicos.”

Nesse aspecto, o planejamento, para uma escola, teria de levar em conta turmas futuras, capacidade dos professores, quantidade de professores, e todas as variáveis que fazem uma escola funcionar.

Apesar do caráter vital dessa atividade, no caso dos campus do IFRN, ela nem sempre é realizada. Como observado pelo professor Marcus quando entrevistado sobre o assunto, a maioria dos campus não realiza um planejamento

relativo ao quadro de professores. E os que realizam, como o campus Zona Norte, não possuem uma ferramenta facilitadora para o processo, além de planilhas digitais.

As tais planilhas digitais eram compostas de várias abas, sendo cada uma destinada para um curso diferente, nelas eram apresentadas todas as disciplinas que compõem a matriz curricular e seus respectivos créditos por semestre que varia devido à quantidade de turmas ativas no período além de informações afins para uma tentativa de facilitar o trabalho de quem o manuseia.

Figura 1 - Planejamento do curso de eletrônica na planilha antiga

The image shows a complex Excel spreadsheet used for course planning. The spreadsheet is organized into columns representing semesters (from 1st to 6th) and rows representing various courses and disciplines. The data is color-coded, with yellow and blue cells indicating specific information. The bottom of the spreadsheet shows a navigation bar with tabs for 'Eletrônica', 'Manutenção', 'Informática', 'Licenciatura Info', 'Comercio', 'CST Marketing', and 'Total'. The spreadsheet is titled 'Planejamento do curso de eletrônica na planilha antiga'.

Fonte: Equipe ACHA

Figura 2 - Planejamento do curso de manutenção na planilha antiga

The image shows a complex spreadsheet with multiple sections. Each section has a red header and contains various data points, including course names, discipline lists, and workload calculations. The data is color-coded with yellow, blue, and red cells. The spreadsheet is organized into columns and rows, with some cells containing text and others containing numerical values. The overall layout is dense and detailed, representing a comprehensive planning tool for course management.

Fonte: Equipe ACHA

Tal planejamento, para os campus do IFRN, englobaria os cursos oferecidos e as turmas que existem e virão a existir, com suas devidas disciplinas (tanto as comuns como as técnicas), e suas respectivas cargas horárias, além dos projetos e outras atividades extrassala que poderiam ser realizadas pelos professores para calcular a carga horária dos docentes. Devendo levar em conta, além disso, o agrupamento de professores, e a interdisciplinaridade destes.

Esse volume de informações, quando agrupado de maneira correta, serve para avaliar a necessidade da contratação de novos professores, prever a possibilidade de oferecer mais ou menos turmas, saber a sobrecarga dos professores e, de forma geral, permitir uma resposta mais rápida do corpo administrativo do campus a problemas relativos a quantidade de docentes.

Nesse sentido, a diretoria acadêmica do campus Zona Norte do IFRN implementa esse planejamento, denominado PCH (Projeção de Carga Horária) através de planilhas digitais, buscando os benefícios que este pode trazer. Entretanto, esse método tem se mostrado pouco eficiente, uma vez que, com o

crescimento do número de turmas e cursos, o volume de dados aumenta de forma a dificultar a percepção de todas as nuances pelo responsável pela operação.

As consequências disso podem ser observadas no dia a dia do campus, quando em certos períodos do ano os professores passam a acumular carga horária acima do que deviam.

Além disso, o IFRN que, como órgão público, necessita que haja transparência nas finanças, o processo de PCH influencia diretamente na burocracia necessária para o funcionamento da escola, como a abertura de editais para contratação de discentes ou a chamada de professores temporários.

1.1 JUSTIFICATIVA

O processo de PCH é feito de forma manual, utilizando um software de criação de planilhas desde o início dessa prática, atrasando assim muito a finalização do processo, tendo em vista que é preciso conciliar a carga horária dos professores com as disciplinas lecionadas, isso mostra que “fazendo uso de planilhas e processo manual, o tempo para inserção dos dados e análise dos resultados é muito maior e suscetível ao erro.” (MBCONTROL, 2014)

Essa tarefa acaba sendo maior porque seus cálculos e complexidade acabam requerendo bastante atenção, fazendo com que mais de um servidor acabe trabalhando nele demandando mais trabalho conseqüentemente. Ele possui uma margem alta para erros pois como todas as fórmulas e cálculos são manuais um número errado pode colocar todo trabalho a perder, sendo sempre necessário ainda um *backup* para não perder todos os dados caso o arquivo se corrompa.

1.2 OBJETIVOS

O principal objetivo do projeto é desenvolver um software que agiliza e automatiza o referido processo, diminuindo a margem de erros, permitindo projeções mais precisas e possibilitando o planejamento de turmas futuras em virtude da disponibilidade dos professores, além disso são objetivos mais específicos:

- Agilizar e facilitar o processo de PCH;
- Promover melhorias na gestão do campus;
- Exportar a tecnologia produzida para outros campus;

- Integrar o sistema produzido a sistemas já existentes;
- Simular a quantidade de turmas por curso que podem ser criadas por período baseado na disponibilidade de professores;
- Facilitar a distribuição de professores nas ofertas disponíveis baseado na carga horária de cada um;
- Organizar as informações de professores, áreas e grupos permitindo ao diretor acadêmico gerenciá-las.

1.3 RESULTADOS ESPERADOS

Com os nossos objetivos atingidos, ou seja, quando tivermos concluído o software proposto, esperamos observar diversos efeitos positivos:

1. Inicialmente, diminuir a carga de trabalho dos responsáveis pelo processo de Projeção de Carga Horária;
2. Também esperamos que em posse do software o instituto seja capaz de melhor distribuir a carga horária dos professores, promovendo melhor qualidade de vida para o corpo docente;
3. Pode se esperar também que, como efeito do bom uso de uma possível projeção de carga horária, haja diminuição do desfalque de professores, melhorando a qualidade do ensino oferecido pelo colégio;
4. E, assumindo uma boa experiência com o software, a aplicação desse mesmo modelo a outros campus do IFRN e outras escolas no estado do Rio Grande do Norte.

1.4 METODOLOGIA

Para o desenvolvimento desse projeto utilizaremos, a estrutura do processo de desenvolvimento de software OpenUp, que é uma implementação do processo unificado. Esse tipo de processo segundo Schach (2009, p.73) “não é uma série específica de etapas que se, forem seguidas, resultam na construção de um produto de software” já que existem diversos tipos de softwares tornando difícil existir uma metodologia que se aplique a todos os casos. Na verdade o processo unificado deve

ser tratado como uma base adaptável que se encaixe ao software em desenvolvimento. Desse modo, o trabalho ficou dividido nas seguintes etapas, conhecidas como iterações:

- **Fase de Concepção:** na qual os objetivos são compreender o que vai ser construído, identificar as funcionalidades do sistema a ser projetado, entender os riscos do projeto e começar a estudar as tecnologias que serão utilizadas, além disso, é nessa fase que definimos se o produto é viável de ser produzido e se é possível entregá-lo no prazo;
- **Fase de Elaboração:** na qual a meta é refinar as necessidades da fase anterior, como melhorar o entendimento dos requisitos do sistema, criar e estabelecer a base da arquitetura do sistema e mitigar os riscos prioritários;
- **Fase de Construção:** na qual o propósito é criar uma versão do software, que seja operacional, que esteja pronto para a transição e para ser utilizado pela comunidade, de modo que ele se inicie numa fase beta e a cada entrega o se aprimore e esteja mais perto da conclusão, dessa forma o usuário poderá utilizar os recursos que são entregues a cada prazo, podendo fazer comentários e dizer se as funcionalidades estão de acordo com o que foi proposto;
- **Fase de Transição:** nessa fase, as finalidades dos objetivos são ajustar as funcionalidades, performance e qualidade do produto da fase de construção, além de também buscar garantir que todas as exigências do cliente ou usuário do software tenham sido atendidas, então, baseado no *feedback*, são feitas as últimas alterações no produto.

1.5 ORGANIZAÇÃO DO DOCUMENTO

Este documento está dividido em cinco capítulos. Neste primeiro é abordada a introdução contendo a justificativa do projeto, objetivos, resultados esperados e a metodologia utilizada. No segundo falaremos da fundamentação teórica onde será explicado mais da parte técnica do projeto. No terceiro é explicada a solução proposta onde terá contido os resultados esperados para o fim do projeto. No quarto

é falado dos resultados obtidos com o desenvolvimento desse projeto. E, por último, no quinto são relatadas as considerações sobre o processo abordado no documento.

2 FUNDAMENTAÇÃO TEÓRICA

Nesse tópico abordaremos as características e informações com relação a parte técnica do sistema, desde a parte mais abstrata e superficial do mundo da informática até o nível mais implementado.

A desmistificação começa principalmente ao que vem a ser um Software. E de acordo com Pressman (2011, p. 31) o software pode ser considerado como um produto e ao mesmo tempo um veículo do produto, assumindo desse modo um papel duplo. No papel de produto ele é responsável por dar um “potencial computacional representado pelo *hardware*” entretanto quando possui um papel de veículo de distribuição do produto ele funciona como tanto como um controlador do *hardware*, como intermédio à comunicação de informação e até como um gerenciador para criação e controle de outros programas.

Outra característica que pode ser atribuída ao produto é que ele se pode ser definido como um Software de Aplicação, já que segundo Pressman (2011, p. 34) trata-se de um sistema que solucionam uma necessidade específica, e aplicações dessa área são responsáveis pelo processamento de dados técnicos que facilito atividades de tomada de decisões administrativas. Além disso, há também as Aplicações Web pode ser definido como: “um conjunto de arquivos de hipertexto interconectados, apresentando informações por meio de texto e informações gráficas limitadas” (PRESSMAN, 2011, p. 35). Através dessa definição, a caracterização do sistema produzido pode ser denominado como uma Aplicação Web.

2.1 ENGENHARIA DE SOFTWARE

Considerando que este documento trata do processo de desenvolvimento de um sistema de informação, que pode ser denominado *software*, é importante conhecer que foi desenvolvido um processo pelo qual a construção do sistema deve passar, levando aos princípios de Engenharia de Software.

Uma definição muito conhecida é dado por Fritz Bauer na primeira conferência relacionada ao assunto, conforme citado por Pressman (2011, p.39):

“Engenharia de software é o estabelecimento e o emprego de sólidos princípios de engenharia de modo a obter software de maneira econômica, que seja confiável e funcione de forma eficiente em máquinas reais”

Porém essa definição se torna empobrecida quando se aplicada de forma mais aprofundada no ramo da informática, e com isso há também a definição proposta pela IEEE, tratando de forma mais abrangente com relação ao assunto:

“Engenharia de software: (1) A aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de software, isto é, a aplicação de engenharia ao software. (2) O estudo de abordagens como definido em (1)” (PRESSMAN, 2011, p. 114)

2.1.1 Planejamento e gerenciamento

Seguindo modelos mais aprofundados na engenharia de software, constata-se a necessidade de um planejamento e de um gerenciamento das atividades que estão e que entrarão em fase de realização, isso para que tudo deva ocorrer em segurança e com qualidade.

Segundo a definição de Eduardo (2008), o Planejamento é referente a um processo contínuo e dinamizado que consistem em um grupo de ações padronizadas, ou seja, há ações intencionadas, ações coordenadas, ações integradas e até orientadas, tudo com um único prol que é a busca da realização de um objetivo, através de adequações a serem seguidas, através de pontos reguladores como o prazo, qualidade, desempenho, segurança, custos e etc.

E Gerenciamento de acordo com Leite (2007), trata-se de cumprir todos os traços que foram definidos no processo de planejamento. Quando referente ao gerenciamento de *software*, é a responsabilização à garantia da qualidade do produto em si.

2.1.2 Processo unificado

Visando a forma de trabalho para o desenvolvimento de *software*, cujo pode ser incluído como um sistema informacional, é necessário a escolha de processos para que haja um trabalho conciso e organizado no ramo da engenharia de software.

E com isso o Processo Unificado (PU) trata-se de um processo popular para desenvolvimento de *software*, que segundo Cleber (2016) visa à construção de sistemas orientados a objetos, utilizando de ciclos de vidas iterativos, esse baseado em refinamentos e incrementos refinadas com fim de uma convergência adequada e agradável do produto desenvolvido.

De acordo com Pressman (2011, p. 71), o PU tem como objetivo intensificar e reconhecer a importância da comunicação entre desenvolvedores e seus clientes, para que esses possam aplicar a visão sobre o sistema. Além disso sua assistência visa manter os produtores focados nas metas coerentes, utilizando de vias como a compreensividade, confiança em mudanças futuras e reutilização.

2.1.3 UML

A *Unified Modeling Language* (linguagem de modelagem unificada), conhecida pela sigla UML, é “uma linguagem-padrão para descrever/documentar projeto de *software*. A UML pode ser usada para visualizar, especificar, construir e documentar os artefatos de um sistema de *software-intensivo*”. (BOOCH, 2005)

2.1.4 MVC

O modelo MVC (*Model, View e Controller*) é, segundo Ramos (2015): “Nada mais que um padrão de arquitetura de software, separando sua aplicação em 3 camadas. A camada de interação do usuário(view), a camada de manipulação dos dados(model) e a camada de controle(controller).”

Ele é dividido em três camadas onde cada uma possui uma função específica, sendo elas:

- **Model:** Que é responsável pela manipulação de dados, tendo seu uso mais associado a gravar e recuperar dados do banco de dados, porém também pode-se realizar ações de consultas e cálculos nesta camada. É nela também

que definimos a estrutura dos objetos e seus atributos e métodos além de dela está ligada ao *controller*.

- **View:** Ela também possui um contato direto com o *controller* e é responsável pela interação com o usuário, ou seja, tudo que o usuário ver é apresentado pela *view*. Os arquivos que ficam nesta camada normalmente são do tipo “.html” ou “.php”.
- **Controller:** Faz a ligação entre o *view* e o *model* e faz a validação de informações do sistema.

A principal vantagem de se utilizar esse padrão é dividir o sistema, que seria o problema, em partes para assim fragmentar também esse problema em menores de modo que se ocorrer um problema em uma parte não compromete todo o sistema.

2.2 SISTEMAS DE APOIO À DECISÃO

Os sistemas atuais abrangem as intenções de quem desenvolve como de quem solicita, desde suprir necessidades de assistência até o desenvolver novas funções no mundo da informática.

Com isso é preciso identificar a principal característica do que está sendo desenvolvido e com isso em mente, o produto fornecido por esse trabalho se encaixa no grupo de Sistema de Apoio à Decisão (SAD).

Segundo a Portogente (2016), um SAD refere-se a um sistema que fornece meios para que se possa gerenciar um negócio em variados níveis de organização. Outra definição mais pertinente diz que o SAD trata-se de um sistema de computação que auxilia decisões de ações a serem tomadas em alguma área organizacional, tudo isso utilizando de meios de análise lógica e quantitativa.

O SAD, de acordo com Barbosa (2008), se diferencia dos demais sistemas de informações devido seu direcionamento, que é voltado a parte de planejamento estratégico das instituições, auxiliando no gerenciamento de dados específicos e organizações rotineiras.

2.3 FRAMEWORK

Segundo Jaques (2016), o *Framework* é uma composição de códigos abstratos e/ou genéricos, muitas vezes em forma de classes, que foi desenvolvido em alguma linguagem de programação, e todo seu relacionamento entre si mesmo disponibiliza ao seu usuário desenvolvedor funcionalidades específicas para facilitar e/ou até atribuir novas características ao sistema.

“Em outras palavras, é como uma caixa de ferramentas, um kit que possui diversas funcionalidades devidamente implementadas, testadas e prontas para serem utilizadas na construção de softwares, poupando ao desenvolvedor tempo e trabalho na elaboração de operações básicas como acesso a banco de dados, sistema de templates, mapeamento de rotas e validação de dados.” (JAQUES, 2016)

2.4 SUMÁRIO

Observou-se que nesse capítulo foram abordados os assuntos necessários para a devida compreensão do documento e para a construção do projeto. Ainda no capítulo 3 será abordada a solução proposta, de forma detalhada para a devida compreensão do processo de desenvolvimento do *software*.

3 SOLUÇÃO PROPOSTA

Esta seção aborda o objeto que deve ser gerado ao fim do projeto e suas características. O ACHA (Assistente de Carga Horária Acadêmica) é uma plataforma para a realização do processo de PCH e controle de carga horária dos docentes do IFRN, baseado no conceito de SAD.

Sua construção obedece uma série de requisitos que foram levantados durante as fases de planejamento do projeto. Esses requisitos, que foram levantados com base em entrevistas com os principais interessados e da análise do processo ao qual ele deve auxiliar, “[...] São as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento” (SOMMERVILLE, 2011, p.57).

3.1 LEVANTAMENTO DE REQUISITOS

Segundo o padrão de Projeto Unificado implementado pelo sistema OpenUp, as primeiras etapas do nosso projeto envolviam o levantamento de requisitos, uma vez que sabendo essas características do produto o desenvolvimento dele é apenas questão de escrita de trazer para a realidade aquilo que foi teorizado.

Nessa etapa realizamos diversas atividades, entre elas destacam-se: compreender a problemática, ou seja, entender como se dá o processo de PCH; identificar os requisitos, elucidando o que um software necessita para proporcionar uma solução para a necessidade que fosse melhor que a anterior; separá-los em funcionais e não funcionais; e, por fim, ordená-los conforme a importância.

A compreensão da problemática se deu através de discussões entre os membros do grupo, com os orientadores e com *stakeholders*. Além disso, uma análise das tabelas de projeção utilizadas anteriormente nos permitiu uma visualização mais concreta do problema.

Com o entendimento do procedimento de supracitado, foi possível destacar as necessidades as quais deveriam ser atendidas pelo sistema cujo objetivo do projeto é desenvolver. Nessa etapa foi necessária, mais uma vez, reuniões entre os

componentes do grupo entre si e com os orientadores, que nos guiaram pelo processo. A principal técnica utilizada foi a de *brainstorming*, a qual é “[...] baseada numa discussão em que várias pessoas fazem muitas sugestões para que sejam escolhidas as melhores ou uma fusão delas.” (PRIBERAM, 2016).

Após elucidados os requisitos, a próxima coisa a ser feita é classificá-los como funcionais e não funcionais. Segundo Sommerville (2011, p. 59) os requisitos funcionais “São declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações.” e os requisitos não funcionais “São restrições aos serviços ou funções oferecidos pelo sistema.”.

A última etapa desta atividade consistiu em apresentar uma ordem de importância para os requisitos. Para isso, foi decidido que os requisitos seriam classificados como importantes ou não. Assim, o desenvolvimento deles seria feito com mais atenção.

Desse modo, os requisitos que representam funcionalidades do sistema e possíveis ações tomadas pelo usuário foram classificados como funcionais e evidentes.

Tabela 1 - Requisitos funcionais

COD	NOME	DESCRIÇÃO	TIPO
F01	Gerenciamento de cursos	Permite a adição, alteração, visualização e remoção de cursos no sistema	Evidente
F02	Gerenciamento de matrizes curriculares	Permite a adição, alteração, visualização e remoção de matrizes curriculares no sistema	Evidente
F03	Gerenciamento de professores	Permite a adição, alteração, visualização e remoção de professores no sistema	Evidente
F04	Gerenciamento de turmas	Permite a adição, alteração, visualização e remoção de	Evidente

		turmas no sistema. Permitindo adicionar uma disciplina com um ou mais professores.	
F05	Gerenciamento de campus	Permite a adição, alteração, visualização e remoção de campus no sistema	Evidente
F06	Exportação de dados	Permite a exportação dos dados referentes a curso, matriz, professores, turmas e campus em formatos xlcx ou xml	Evidente
F07	Importação de dados	Permite a importação dos dados referentes a curso, matriz, professores, turmas e/ou campus em formatos xml	Evidente
F08	Gerenciamento de planejamentos	Permite a adição, alteração e remoção de planejamentos de oferta de turmas por semestre no sistema.	Evidente
F09	Visualização de oferta	Permite visualizar a viabilidade da oferta planejada, em relação ao crédito dos professores, informando a disponibilidade desses e/ou a necessidade de chamar novos professores	Evidente
F10	Definição de coordenadores de cursos	Permite definir professores, relacionados as áreas técnicas, como coordenadores dos cursos nos quais eles lecionam.	Evidente
F11	Gerenciamento de áreas	Permite a adição, alteração, visualização e remoção de áreas de disciplinas no sistema, assim como associar as disciplinas às áreas que elas pertencem.	Evidente

F12	Associação de disciplinas	Permite associar os professores à(s) disciplina(s) que ele leciona. (As disciplinas podem ter vários professores)	Evidente
F13	Execução de planejamento	Permite definir um planejamento já configurado, e possível de ser executado, como ativo, para que ele seja o modelo de gerenciamento acadêmico para os próximos semestres.	Evidente
F14	Distribuição de professores	Permite que os coordenadores de curso, já definidos pelo diretor acadêmico, distribuam os professores para um período, também já definido pelo diretor acadêmico.	Evidente
F15	Agrupamento de professores	Permite agrupar professores por características similares nas disciplinas em que eles lecionam (ex.: Informática, Linguagem, Natureza, etc).	Evidente
F16	Gerenciamento de disciplinas	Permite a adição, alteração, visualização e remoção de disciplinas no sistema.	Evidente
F17	Visualização disciplinas por semestre	Permite visualizar as disciplinas que serão lecionadas em determinado semestre.	Evidente
F18	Associação de grupos as áreas	Permite associar os grupos de professores às áreas por eles lecionadas	Evidente
F19	Filtragem de dados	Permite que os dados visualizados, nas telas com muita informação, possam ser filtrados, a fim de facilitar o entendimento	Evidente

Fonte: Equipe ACHA

Os requisitos que representam características gerais do sistema, foram ditos não funcionais e classificados como opcionais ou obrigatórios.

Tabela 2 - Requisitos não funcionais

COD	NOME	DESCRIÇÃO	TIPO
NF01	Autenticação	Autenticação do usuário utilizando session	Obrigatório
NF02	Responsividade	Adaptabilidade da página a diferentes dispositivos	Opcional
NF03	Design intuitivo	Torna mais intuitiva a utilização da página pelo usuário	Obrigatório
NF04	Acessibilidade	Design que permita acessibilidade	Obrigatório
NF05	Banco de dados	MySql	Obrigatório
NF06	Linguagens de desenvolvimento	php, javascript, html e css	Obrigatório
NF07	Browser	Chrome	Opcional
NF08	Framework	Bootstrap	Opcional
NF09	Arquitetura da informação	Organização do conteúdo do sistema ao usuário	Obrigatório

Fonte: Equipe ACHA

3.2 MODELAGEM DO SISTEMA

Este tópico descreve em detalhes o processo de modelagem do sistema, abordando os casos de uso e suas descrições, além de abordar a arquitetura do sistema.

3.2.1 Casos de uso

Um dos conceitos-chave quando se fala de Engenharia de *Software* é o de caso de uso. “Essencialmente, um caso de uso conta uma história estilizada sobre como um usuário final (...) interage com o sistema sob um conjunto de circunstâncias específicas.” (PRESSMAN, 2011, p. 137).

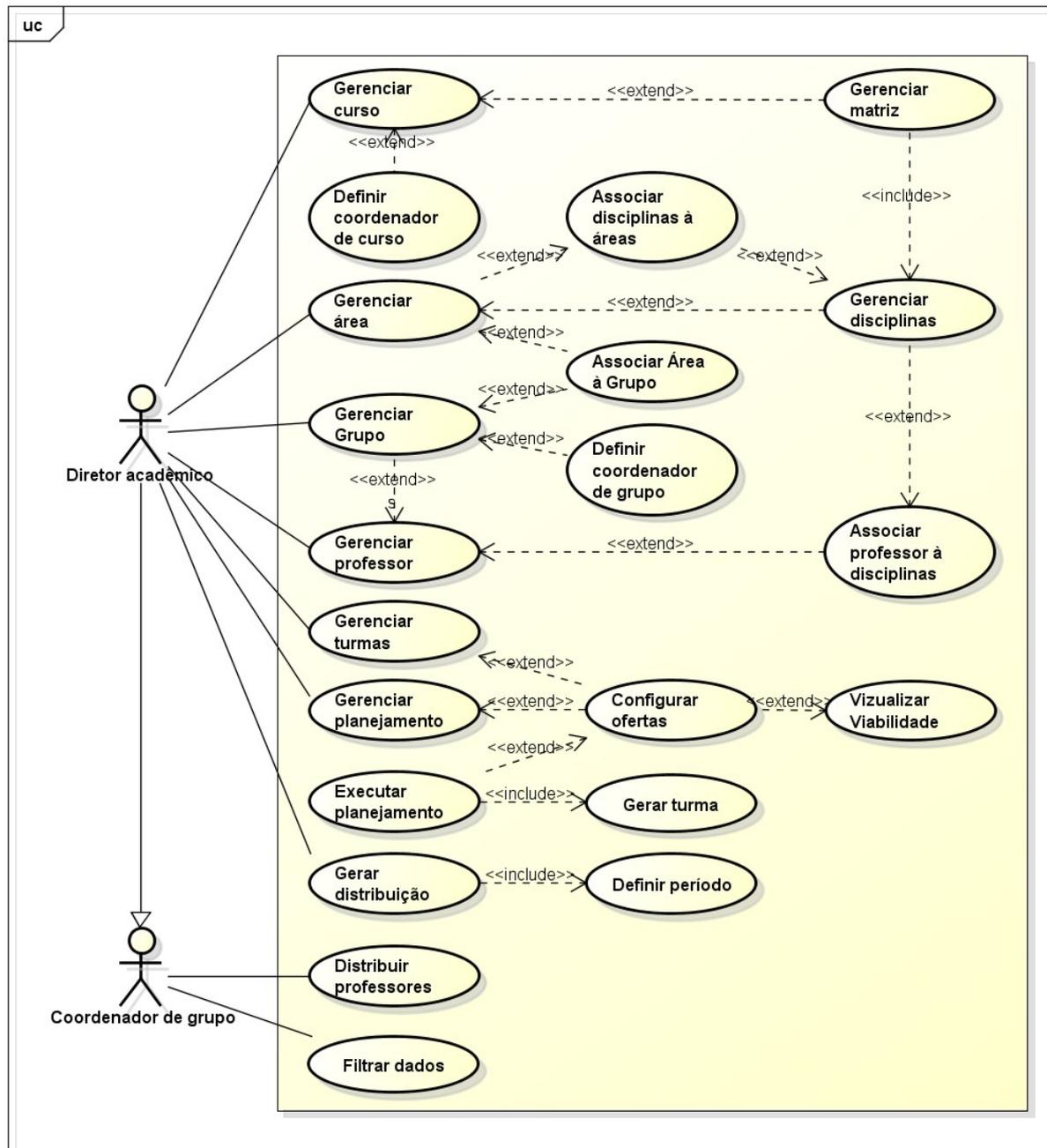
Sob esta ótica, estes foram inferidos pelo grupo com base nos requisitos elucidados anteriormente e através da dedução de possíveis atividades realizadas pelos usuários do sistema, incluindo os diferentes níveis de acesso. Para a construção dos mesmos, foi utilizada uma ferramenta para a criação de diagramas de *software*, o Astah Community.

Devido a natureza da aplicação, observa-se apenas um módulo. Isso se deve ao fato de todas as atividades do sistema possuírem influência uma sobre a outra, e dele funcionar de forma uniforme.

- UC1 - Definir coordenador de curso: O diretor acadêmico pode especificar coordenadores para cada curso;
- UC2 - Gerenciar curso: O diretor acadêmico pode criar um novo curso e visualizar, modificar – isto é, editá-lo e/ou associá-lo a uma matriz - ou remover os já existentes;
- UC3 - Gerenciar área: O diretor acadêmico pode criar novas áreas de educação, visualizá-las, modificá-las - isto é, configurar as disciplinas contidas nessas áreas - ou apagá-las;
- UC4 - Gerenciar grupo: O diretor acadêmico pode criar grupos de áreas, visualizar, editar, alterar e/ou organizar as áreas associadas ou apagar grupos já existentes;
- UC5 - Gerenciar professor: O diretor acadêmico pode cadastrar, dentro da aplicação, professores, visualizar, editar ou apagar professores existentes;
- UC6 - Criar planejamento (maior risco): O diretor acadêmico pode criar um planejamento de oferta de turmas para os próximos períodos.
- UC7 - Abrir planejamento: O diretor acadêmico pode abrir um planejamento de oferta existente no sistema;
- UC8 - Executar planejamento: O diretor acadêmico pode executar, isto é, oficializar um planejamento de oferta existente;

- UC9 - Gerar distribuição: O diretor acadêmico pode validar a distribuição das matérias, por curso, geradas pelo sistema;
- UC10 - Distribuir professores: O coordenador de grupo pode distribuir os professores de acordo com os créditos, por disciplina, requeridos;
- UC11 - Associar disciplinas às áreas: O diretor acadêmico pode associar um professor a uma ou mais disciplinas;
- UC12 - Associar áreas a grupo: O diretor acadêmico pode associar uma ou várias áreas à um grupo;
- UC13 - Configurar ofertas: O diretor acadêmico pode distribuir as ofertas desejadas no decorrer dos períodos e ser informado caso o planejamento seja viável ou não, isto é, que a quantidade de professores no campus supere a demanda requisitada;
- UC14 - Gerar turma: Com base nos parâmetros no planejamento, o sistema irá gerar as futuras turmas;
- UC15 - Definir período: Após gerar a distribuição, o diretor acadêmico deve definir o período que o coordenador de grupo poderá distribuir os professores;
- UC16 - Definir coordenador do grupo: O diretor acadêmico pode especificar coordenadores para cada grupo;
- UC17 - Gerenciar matriz: O diretor acadêmico pode criar uma nova matriz curricular, visualizar, modificar – isto é, gerenciar os núcleos da matriz e atribuir disciplinas a esses – ou apagar matrizes já existentes;
- UC18 - Gerenciar disciplinas: O diretor acadêmico pode criar disciplinas – associando-as a áreas de educação, visualizar, modificar ou apagar as disciplinas já existentes;
- UC19 - Associar professor às disciplinas: O diretor acadêmico pode associar um professor a uma ou mais disciplinas;

Figura 3 - Casos de uso do sistema



Fonte: Equipe ACHA

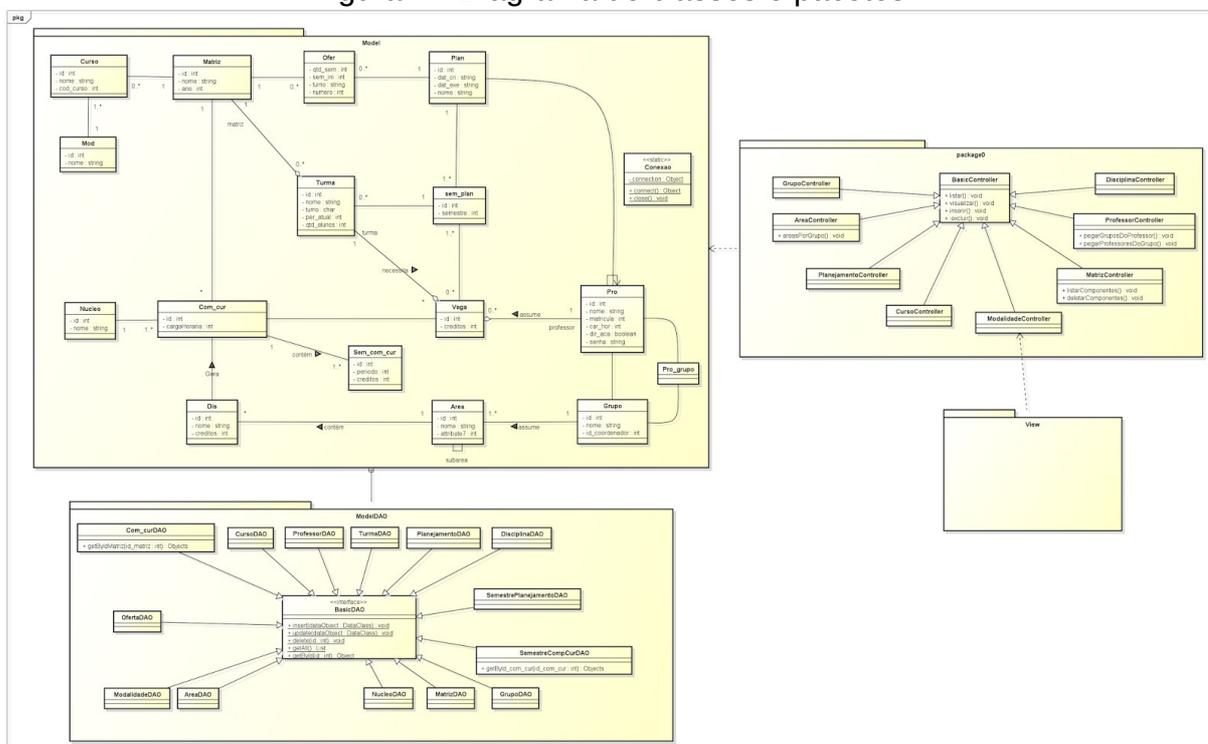
- UC20 - Visualizar viabilidade (maior risco): Simulação para análise das possibilidades de se desenvolver um novo planejamento;
- UC21 - Gerenciar turmas: O diretor acadêmico pode criar uma nova turma e pode editar, visualizar, modificar, juntar, separar e agrupar uma ou várias turmas;

- UC22 - Filtrar dados: Enquanto estiver em janelas com diversos dados, o usuário poderá escolher os filtros desejados para facilitar a visualização da informação.

3.2.2 Arquitetura do sistema

Para descrever o sistema, escolhemos utilizar o padrão de arquitetura MVC (Model, View, Control). Seguindo esse conceito, foram produzidos o diagrama de classes que são usados no desenvolvimento de um modelo de sistema orientado a objetos para mostrar as classes de um sistema.

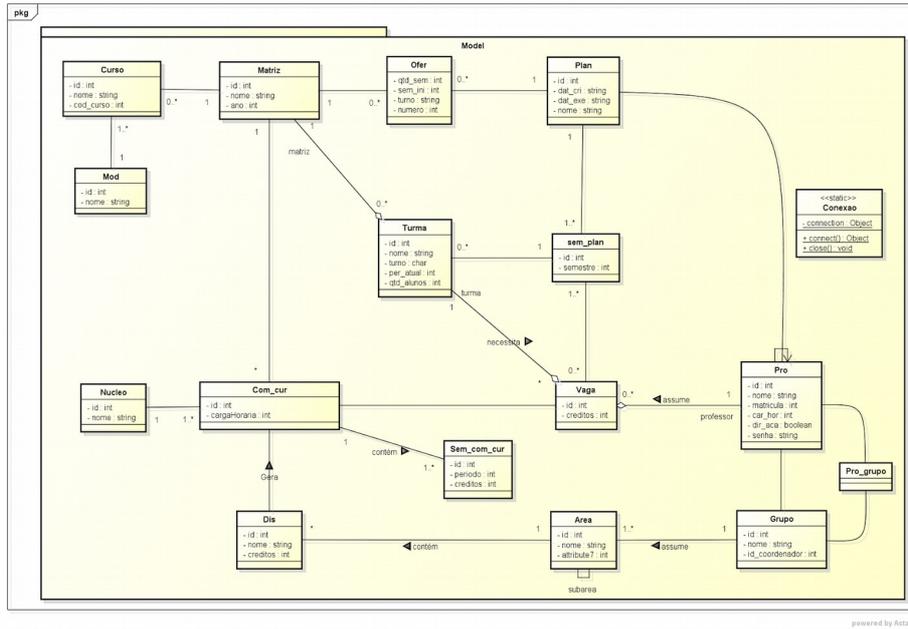
Figura 4 - Diagrama de classes e pacotes



Fonte: Equipe ACHA

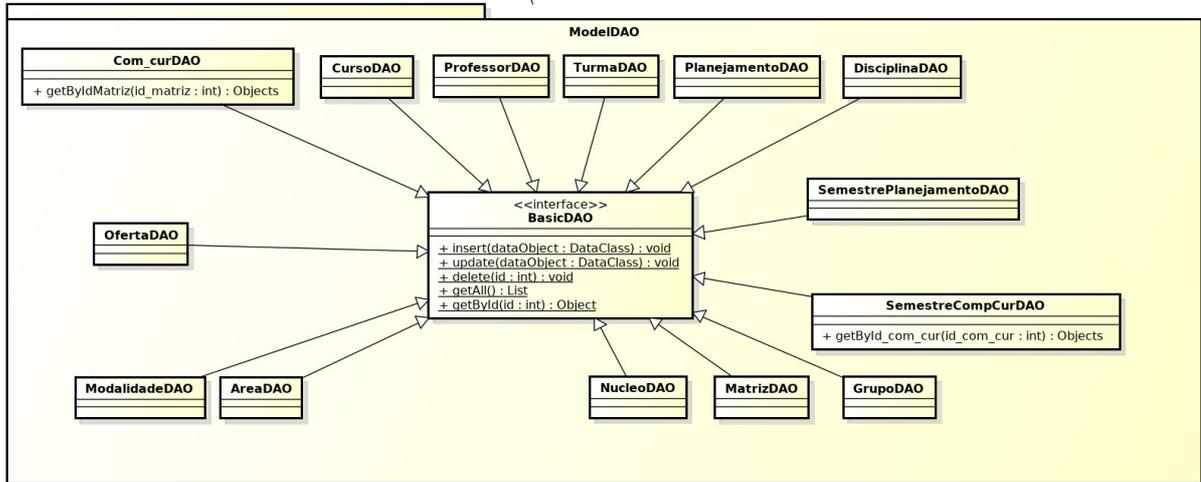
Observa-se nesse diagrama a presença das três camadas do MVC. O *model*, que trata da representação dos dados; o *view*, que trata da amostragem dos dados e o *control*, que trata do controle da aplicação. A camada de *model*, nesse caso, ficou dividida em dois pacotes, devido ao padrão de arquitetura adotado. O primeiro apresenta as classes de VO (*value object*), responsáveis pela representação em si, e as classes de DAO (*data access object*), responsáveis pelo acesso ao banco de dados.

Figura 5 - Pacote model



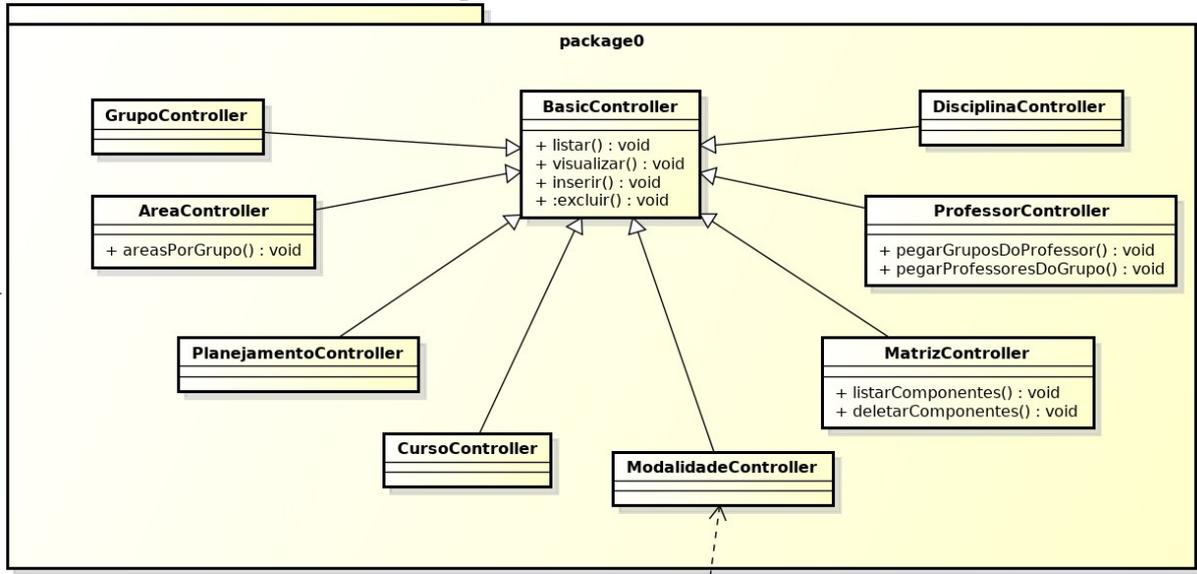
Fonte: Equipe ACHA

Figura 6 - Pacote ModelDAO



Fonte: Equipe ACHA

Figura 7 - Pacote control



Fonte: Equipe ACHA

3.2.3 Protótipos de baixa fidelidade

Os protótipos de baixa fidelidade, tem como intenção definir um esboço do que deva ser seguido pela equipe produtora, para que não olvide da real finalidade do produto e com isso haja uma maior facilidade na produção sem se desprender do que se deseja.

Figura 8 - Tela inicial



Fonte: Equipe ACHA

A imagem anterior apresenta uma ideia de como deve ser a tela inicial do sistema, onde ela apresenta as turmas existentes divididas entre os cursos aos quais elas pertencem. Já a próxima figura exibe a tela do planejamento.

Figura 9 - Tela de planejamento

Fonte: Equipe ACHA

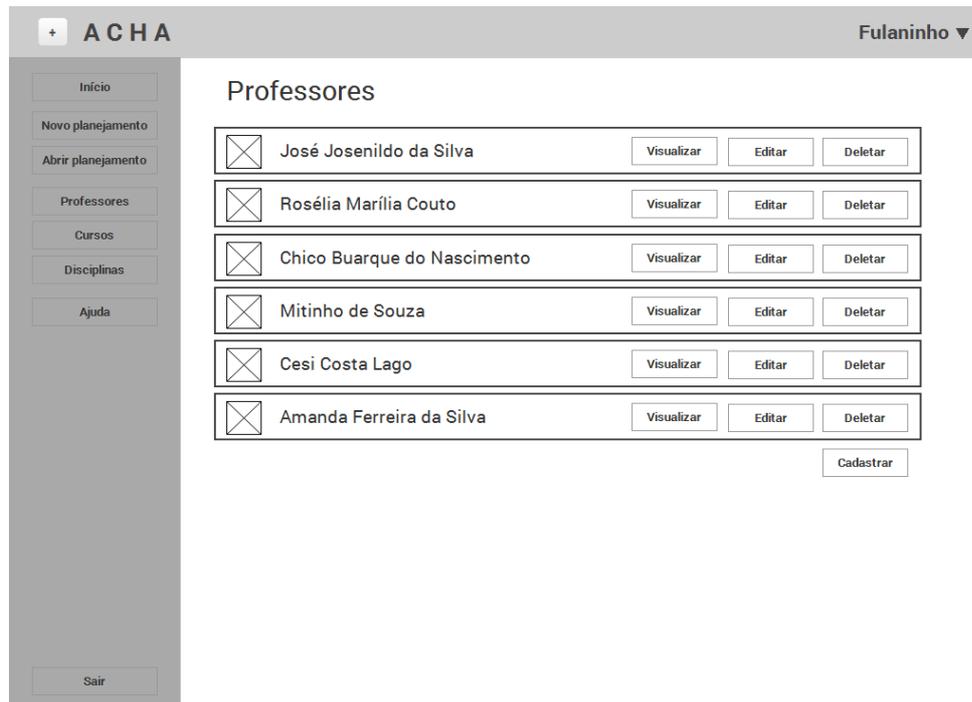
Figura 10 - Tela disciplinas

Disciplina	Área	Créditos
Programação Web	Informática	16
Redes de computadores	Informática	22
Projeto de desenvolvimento de software		26

Fonte: Equipe ACHA

A figura anterior mostra uma ideia de como deveria ser um CRUD para disciplinas. Já a figura que vem a seguir trata de um crud para professores.

Figura 11 - Tela professores

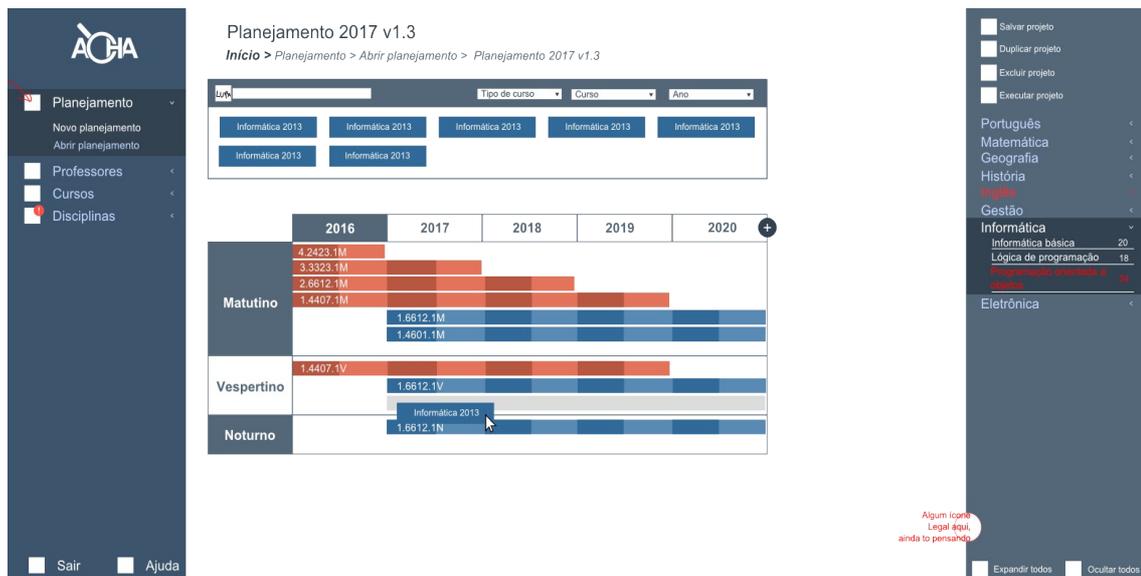


Fonte: Equipe ACHA

3.2.4 Protótipos de alta fidelidade

Já os protótipos de alta fidelidade ajudam a definir, além do formato das telas, uma identidade visual do sistema, ficando bem próximos do resultado final. No caso exposto a seguir encontra-se a tela de planejamento em alta fidelidade.

Figura 12 - Tela de alta fidelidade para planejamento



Fonte: Equipe ACHA

3.3 MODELAGEM DE BANCO DE DADOS

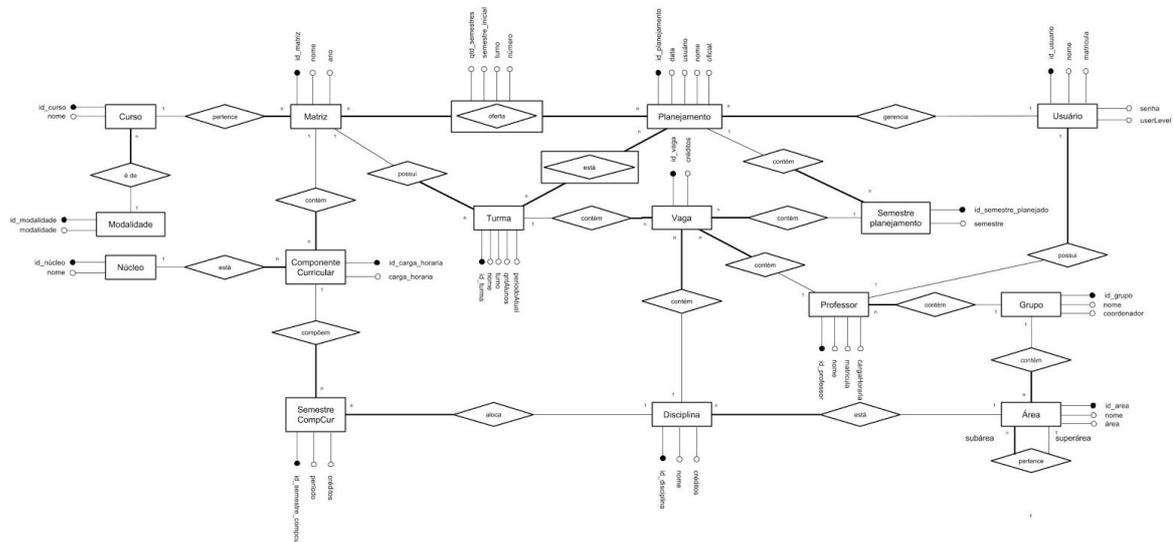
Baseado no modelo do sistema apresentado anteriormente, foi desenvolvido um modelo para o banco de dados. Segundo Elmasri e Ramez (2005, p. 4) “Um banco de dados é uma coleção de dados relacionados. Os dados são fatos que podem ser gravados e que possuem um significado implícito.”

Nessa etapa, tivemos que analisar o modo mais eficaz de armazenar os diversos dados presentes no processo de PCH, como matrizes, disciplinas, professores, áreas, grupos, etc. Desse modo, teríamos um banco de dados, ou ao menos um modelo para um, que representaria estas informações de modo coeso, evitando a duplicidade. Assim, toda essa informação foi transformada em relações, ou seja, as informações seriam construídas através dos relacionamentos entre elas.

Assim, criamos, inicialmente, um modelo entidade-relacionamento do banco de dados, para nos guiar durante o processo de desenvolvimento deste. Segundo Rodrigues (2016):

“O Modelo Entidade Relacionamento (também chamado Modelo ER, ou simplesmente MER), como o nome sugere, é um modelo conceitual utilizado na Engenharia de Software para descrever os objetos (entidades) envolvidos em um domínio de negócios, com suas características (atributos) e como elas se relacionam entre si (relacionamentos).”

Figura 13 - Modelo ER



Fonte: Equipe ACHA

browser o modo como a página deve ser renderizada, ou seja, controlar o aspecto visual dos componentes. Para facilitar a estilização do sistema, utilizamos um *framework* chamado *Bootstrap*, que contém diversos elementos pré-estilizados para facilitar a construção de páginas de modo responsivo.

Para controlar a interação dos usuários com as páginas web, e vice-versa, utiliza-se a linguagem JavaScript. Ela “É uma linguagem de script multi-paradigma, baseada em protótipo que é dinâmica, e suporta estilos de programação orientado a objetos, imperativo e funcional.” (JAVASCRIPT, 2016).

No caso do sistema descrito neste trabalho, o JavaScript é fortalecido por uma biblioteca, o JQuery - que facilita a manipulação de HTML com JavaScript - e por um *framework*, o *AngularJs* - cuja função é facilitar a dinamicidade em páginas web do lado do cliente.

PHP, acrônimo para *PHP: HyperText Processor*, “[...] é uma linguagem interpretada livre, usada originalmente apenas para o desenvolvimento de aplicações presentes e atuantes no lado do servidor, capazes de gerar conteúdo dinâmico na World Wide Web.” Com ele foi possível realizar a comunicação do sistema com o banco de dados.

Para o banco de dados, utilizamos o SGBD (sistema de gerenciamento de banco de dados) MySQL, que utiliza a linguagem de consultas estruturada SQL (*Structured Query Language*) como interface. Nele foram apresentados os dados necessários em formas de tabelas. Para facilitar a utilização desse SGBD, utilizamos a interface gráfica MySQL Workbench, que nos permite utilizar o banco sem escrever códigos SQL, facilita a exportação e edição de tabelas, e criação de diagramas.

Para hospedar tudo isso utilizamos o servidor apache, que é um dos servidores web livres mais utilizados no mundo. Ele nos permite utilizar o PHP, uma vez que este não é uma linguagem interpretada no *browser*.

Já o Astah Community é um *software* gratuito para implementação de UML (*Unified Modeling Language*), a qual é fortemente utilizada nos processos de

engenharia de *software*, como na criação de diagramas diversos. Com ele foram criados os diagramas de caso de uso e de classes.

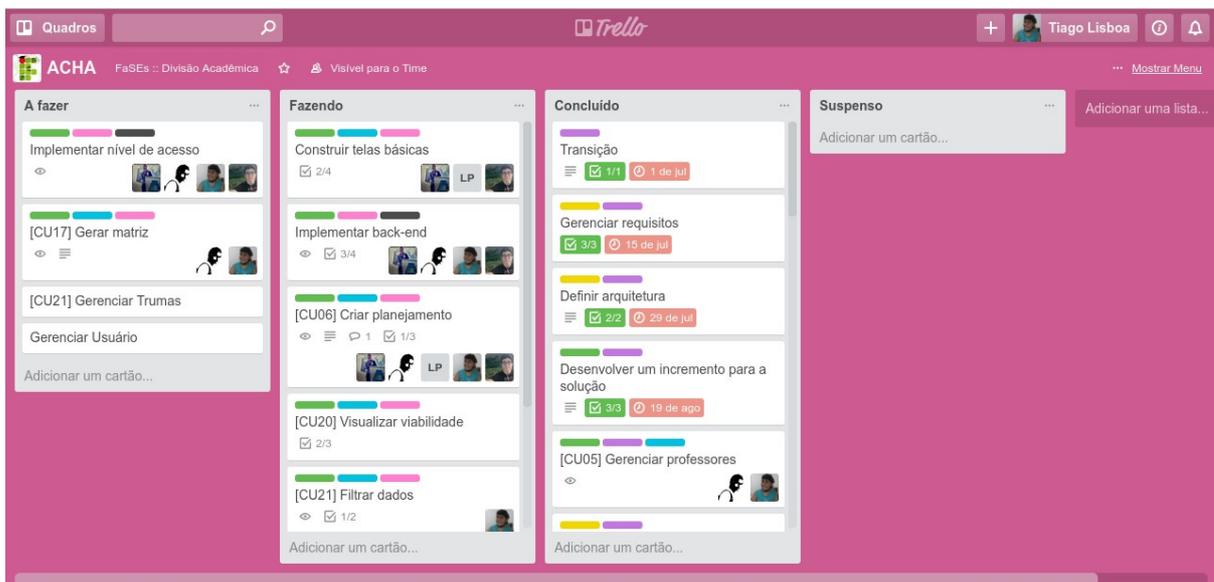
3.5 ORGANIZAÇÃO DO GRUPO

Para marcar as atividades realizadas e separá-las entre os membros da equipe foi utilizado o Trello. Segundo (ILVA, 2016) “O Trello é um sistema de quadro virtual para gerenciamento de tarefas que segue o método "kanban", muito usado no desenvolvimento com Scrum”. (SILVA, 2016). Segundo Vieira (2014) “[...] o Scrum é um framework para organizar e gerenciar trabalhos complexos, tal como projetos de desenvolvimento de software”. E um kanbar é um cartão que controla os fluxos de trabalho em uma equipe.

Assim, foi possível, através dele, organizar o fluxo de trabalho em quatro categorias: a fazer, quando ela é prevista, mas não foi inicializada; fazendo: quando alguém iniciou essa atividade; feita: quando ela já está concluída; e suspensa, quando o grupo desistiu dela.

Desse modo, as atividades foram distribuídas em cartões, com base nos casos de uso elegidos. E estas, para facilitar seu desenvolvimento, foram divididas em tarefas menores, utilizando a ferramenta de checkbox, a qual permite averiguar o progresso do cartão.

Figura 15 - Trello da equipe



Fonte: Equipe ACHA

E para gerenciar, armazenar e compartilhar os códigos utilizamos o git. Segundo Schmitz (2015):

“Git é um sistema de controle de versão de arquivos. Através deles podemos desenvolver projetos na qual diversas pessoas podem contribuir simultaneamente no mesmo, editando e criando novos arquivos e permitindo que os mesmos possam existir sem o risco de suas alterações serem sobrescritas.”

Ele possibilitou ao grupo manter um histórico de alterações no código, facilitando, o regresso a versões anteriores. Para hospedar e compartilhar o repositório, foi utilizado o GitLab, o qual é um servidor gratuito para repositórios git.

3.6 SUMÁRIO

Observou-se que nesse capítulo foram abordados os aspectos da construção do SAD ACHA, como as técnicas de engenharia de *software* utilizadas, os modelos criados, as ferramentas utilizadas e a dinâmica do grupo. Já o próximo capítulo abordará o artefato produzido em si, com foco nas suas características.

4 RESULTADOS OBTIDOS

O ACHA encontra-se com os seus principais requisitos implementados para a entrega da disciplina a qual ele seguia, Projeto de Desenvolvimento de Software (PDS), já que essa concluiu a iteração de Construção e passa para o processo Transição, onde já ocorre a implementação do sistema já para uso do nosso cliente. Entretanto o sistema se encontra aberto a atualizações futuras, de forma que haja a finalização e incrementação de algumas funcionalidades, porém baseado em todos os testes de simulação ele cumpriu com o que estava sendo esperado pela equipe e cumpre os principais requisitos para atividade no campus.

Baseado no que foi proposto no início do projeto, o produto progrediu muito, obtendo uma semelhança alta com o que foi pedido pelo cliente e, além disso, ele apresenta funcionalidades extras como a interatividade na utilização das matrículas dos professores como *logins*.

A principal função almejada era a de criação de planejamento, pois ela define a carga horária que um curso demandará, o ACHA nesse quesito atingiu totalmente a expectativa, de modo que atingindo o principal requisito e vendo o sistema como um todo a semelhança com o que foi proposto é muito grande.

4.1 FUNCIONALIDADES DO SISTEMA

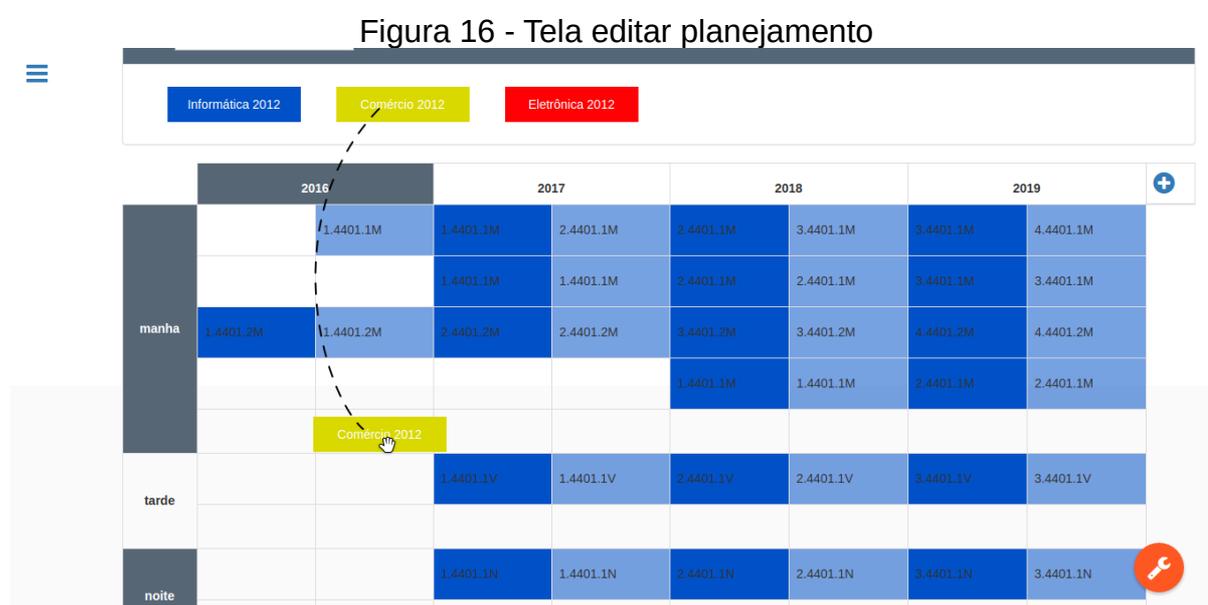
Nesta etapa do processo de desenvolvimento do SAD, pode-se dizer que o ACHA, mesmo que em alguns aspectos ele se afaste do proposto, já trata-se de um sistema capaz de resolver o problema proposto. A seguir, serão apresentados algumas das funcionalidades do sistema.

4.1.1 Planejamentos e Viabilidade

Na parte de planejamentos, o cliente consegue esboçar os cursos aplicados por anos, e com isso tendo uma visão de sua duração, o período inicial e final por semestre, a divisão entre turno matutino, vespertino e noturno. Tal processo funciona através de uma aplicação de arrastamento de item, como pode ser observado na figura a seguir, onde você seleciona o curso desejado e com isso aplica ao ano que deseja iniciar.

Nessa sessão é possível trabalhar com alguns dos requisitos funcionais requeridos pelo sistema, como gerenciamento de planejamento (F08), visualização disciplinas por semestre (F17), visualização de oferta (F09) e execução de planejamento (F13).

O seu funcionamento depende da existência de dados já cadastrados sobre cursos e matrizes, além de áreas e disciplinas e grupos e professores. Os cursos e matrizes geram os cartões coloridos que podem ser arrastados dentro de uma matriz de semestres e turnos. Quando um desses cartões é jogado na tabela é gerada uma turma, que se espalha pelo período de duração da matriz.



Fonte: Equipe ACHA

Com o planejamento em aplicação é possível observar a viabilidade do que se deseja e se será exequível aplicá-lo na escola, tudo ao processo de compatibilidade, que é definida no próprio sistema, com as informações referentes a outras entidades que constituem o instituto, tudo a partir de suas configurações com relações a carga horária disponíveis por profissionais e disciplinas.

Isso é possível pois as turmas são reflexos das matrizes cadastradas e que possuem informações sobre as cargas horárias necessárias para uma turma do curso específico. Essas informações, no planejamento, são agrupadas e somadas, gerando uma estatística concisa de acordo com o que o usuário montou no planejamento. Tudo isso funciona de acordo com os casos de uso: criar

planejamento (UC6), abrir planejamento (UC7), configurar ofertas (UC13), visualizar viabilidade (UC20), gerenciar turmas (UC21), executar planejamento (UC8) e gerar turma (UC14).

Figura 17 - Tela Viabilidade

Autoria Web	0	0	0	0	8	16	16	8
Projeto de Desenvolvimento de Software	0	0	0	0	0	0	8	16
Programação com Acesso a Banco de Dados	0	0	0	0	12	28	28	16
Programação Estruturada e Orientada a Objetos	8	16	40	56	50	26	6	6
Organização e Manutenção de Computadores	0	0	0	0	16	32	32	16
Fundamentos de Lógica e Algoritmos	8	20	20	12	2	3	0	0
Informática	12	12	12	0	3	0	0	0
Códigos e Linguagem	36/0	72/0	108/0	108/0	105/0	93/0	73/0	73/0
Português e Literatura	24	48	72	72	78	78	70	62
Inglês	12	24	36	36	27	15	3	3
Espanhol	0	0	0	0	0	0	0	0
Educação Física	0	0	0	0	0	0	0	0
Qualidade de Vida e Trabalho	0	0	0	0	0	0	0	8
Plaidas Ruins	/62	/62	/62	/62	/62	/62	/62	/62
Ciências Humanas e Sociais	24/0	56/0	80/0	80/0	70/0	64/0	62/0	70/0
História	0	0	0	0	8	16	32	40
Filosofia	8	8	8	8	18	16	8	2
Filosofia, Ciência e Tecnologia	0	0	0	0	0	0	0	8
Sociologia do Trabalho	0	0	0	0	0	0	8	8

Fonte: Equipe ACHA

Observa-se, desse modo, que o trabalho do usuário, que antes tinha de cadastrar individualmente cada valor de carga horária para cada semestre de cada disciplina em uma tabela foi resumido a arrastar e soltar, deixando que o sistema calcule os valores necessários.

Figura 18 - Planejamento antigo

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	
7	Regular	4.4506.1V	4.4506.1V																		Inglês	1
8	Chromium	2.4506.1V	2.4506.1V	3.4506.1V	3.4506.1V	4.4506.1V	Artes	1														
9	Regular nova	1.4506.1M	1.4506.1M	2.4506.1M	2.4506.1M	3.4506.1M	Educação Física	1														
10	Regular nova		1.4506.1V	1.4506.1V	2.4506.1V	Geografia	1															
11	Regular nova			1.4506.1M	Matemática	1																
12	Regular nova							1.4506.1V	Física	1												
13	Regular nova																				Química	1
14	EJA	8.4510.1N																			Biologia	1
15	EJA	7.4510.1V	8.4510.1V																		Informática	1
16	EJA	6.4510.1N	7.4510.1N	8.4510.1N																	Desenho	1
17	EJA	5.4510.1V	6.4510.1V	7.4510.1V	8.4510.1V																Orientação Educacional	1
18	EJA	4.4510.1N	5.4510.1N	6.4510.1N	7.4510.1N	8.4510.1N															Língua Portuguesa	2
19	EJA - novo	3.4510.1V	4.4510.1V	5.4510.1V	6.4510.1V	7.4510.1V	8.4510.1V														Inglês	2
20	EJA - novo	2.4510.1N	3.4510.1N	4.4510.1N	5.4510.1N	6.4510.1N	7.4510.1N	8.4510.1N													Educação Física	2
21	EJA - novo	1.4510.1V	2.4510.1V	3.4510.1V	4.4510.1V	5.4510.1V	6.4510.1V	7.4510.1V	8.4510.1V												Geografia	2
22	EJA - novo			1.4510.1N	2.4510.1N	3.4510.1N	4.4510.1N	5.4510.1N	6.4510.1N	7.4510.1N	8.4510.1N										Matemática	2
23	EJA - novo					1.4510.1N	2.4510.1N	3.4510.1N	4.4510.1N	5.4510.1N	6.4510.1N	7.4510.1N	8.4510.1N								Física	2
24	EJA - novo																				Química	2
25	EJA - novo																				Biologia	2
26	EJA - novo																				Informática	2
27	EJA - novo																				Desenho	2
28	EJA - novo																				Orientação Educacional	2
29	EJA - novo																				Língua Portuguesa	2
30	EJA - novo																				Inglês	3
31	EJA - novo																				Artes	3
32	Português	10	10	9	9	11	11	11	11	11	11	8	8	5	5	5	5	5	5	5	Educação Física	2
33	Inglês	2	2	3	3	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	3	3
34	Artes	2	2	2	2	4	4	4	4	4	4	2	2	2	2	2	2	2	2	2	0	0
35	Educação Física	4	4	6	6	6	6	6	6	6	6	4	4	4	4	4	4	4	4	4	0	0
36	Matemática	8	8	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	0	0
37	Geografia	8	8	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	0	0
38	História	8	8	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	0	0
39	Química	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	3	3
40	Física	2	2	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	0	0
41	Química	10	10	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	4	4
42	Biologia	2	2	3	3	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	4	4
43	Desenho	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
44	Artes	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
45	Sociologia	6	2	4	4	2	6	4	6	4	6	4	4	4	4	4	4	4	4	4	2	2
46	Filosofia	0	4	2	4	2	4	2	4	2	4	2	4	2	2	2	2	2	2	2	0	0
47	Espanhol	3	3	6	6	3	6	3	6	3	6	3	6	3	6	3	6	3	6	3	6	3
48	Educação	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
49	Gestão	31	31	32	32	30	30	30	30	30	30	28	28	28	18	18	9	9	9	9	9	9

Fonte: Equipe ACHA

4.1.2 Matrizes

As matrizes são responsáveis por definir a estrutura que irá compor determinados cursos e o que ele poderá oferecer ao campus, com relação a disciplinas por carga horária, estrutura de ensino e segmento a ser seguido por semestres.

Como pode-se observar na figura a seguir, o cliente do sistema define a distribuição por semestre, para que seja compatível com o disponível por cada disciplina, caso haja uma excedência ou uma escassez do que é disponibilizado o próprio sistema alerta e impede que tal matriz seja computada, cumprindo, desse modo, o requisito funcional de gerenciamento de matrizes curriculares (F02) e funcionando de acordo com o requisito gerenciar matriz (UC17).

Figura 19 - Matriz em construção

Componente	Carga Horária		Núcleo	Semestres								
	Hora/Aula	Horas		1	2	3	4	5	6	7	8	
Português e Literatura	440	330	Técnico	3	3	3	3	3	3	2	2	
Inglês	240	180	Técnico	3	3	3	3	1	0	4	0	
Biologia	280	210	Técnico	3	3	4	4					
Arte	120	90	Técnico	0	2	0	2	0	0	0	0	
Eletrônica Analógica e Digital	160	120	Técnico	2	2	2	2	0	0	0	0	
Geografia	240	180	Técnico	4	4	2	2	0	0	0	0	
História	240	180	Técnico	0	0	0	0	2	2	4	4	

Fonte: Equipe ACHA

4.1.3 Professores

Os dados de professores dentro do software, tanto funciona como uma entidade quanto usuário, para ter acesso aos seus próprios dados. Cada um tem sua matrícula, senha, definição de quanto pode disponibilizar de carga horária, os grupos o qual integram e se o cargo é além de professor, cumprindo, desse modo, os requisitos funcionais de gerenciamento de professores (F03) e agrupamento de professores (F15) e funcionando de acordo com os casos de uso de gerenciar professor (UC5) e associar professor a disciplina (UC19).

Figura 20 - Editar Professor

Alison Hedigiranes

[Início](#) / [Professores](#) / [Alison Hedigiranes](#)

ID: 6

Nome: Alison Hedigiranes

Matrícula: 123

Senha: [Redefinir Senha](#)

Carga Horária: 100%

Grupos:

- Informática
- Códigos e Linguagem
- Piadas
- Ciências Humanas e Sociais
- Ciências da Natureza
- Matemática
- Eletrônica
- Comércio
- Grupo Legal

Cargo: Professor

[Submeter](#) [VOLTAR](#)

Fonte: Equipe ACHA

4.1.4 Grupos, áreas e disciplinas

Tais entidades são compostas por relações entre si, onde uma está inserida na outra, na seguinte sequência: Disciplina faz parte de Área e Área faz parte de Grupos, com isso as relações são dadas por agrupamento.

A disciplina consiste de várias unidades responsável pelo conteúdo que será inserido dentro uma matriz curricular, onde dentro da desta poderá pertencer à algum dos seguintes núcleos: Articulador, Estruturante ou Tecnológico. Além disso é preciso definir a carga horária que será disponibilizada por essa unidade para que seja devidamente distribuída na estruturação de qualquer matriz pelos semestres.

Já área representa uma estrutura que agrega disciplinas por semelhança, onde haverá mais facilidade de organizar as tais em agrupações, onde haja semelhanças com relação ao nível de ensino entre elas, que são aplicados dentro de algum curso.

Já os grupos eles são uma entidade acima de área, tendo como função tanto agrupar todas as áreas e professores que irão compor-la para com isso poder

visualizar a disponibilidade para o lecionamento de tais disciplinas aplicadas em determinadas matrizes quando visualizadas em um planejamento.

Nessas partes que integram o sistema há uma grande gama dos requisitos funcionais que compõem o produto desenvolvido, entre eles estão o gerenciamento de áreas (F11), gerenciamento de disciplinas (F16), associação de disciplinas (F12) e associação de grupos às áreas (F18) e tendo seus funcionamentos de acordo com os casos de uso de gerenciar grupo (UC4), gerenciar área (UC3), gerenciar disciplinas (UC18), definir coordenador de grupo (UC16), associar área à grupo (UC12) e associar disciplina à área (UC11).

Figura 21 - Editar grupo

A interface de usuário para editar o grupo 'Artes'. No topo, há um menu hambúrguer e o título 'Artes'. Abaixo dele, uma barra de navegação mostra 'Início / Grupos / Artes'. O formulário contém os seguintes campos:

- ID: 3
- Nome: Artes
- Coordenador: Fábio Santos

Abaixo do formulário, há uma seção intitulada 'Professores' com uma lista de nomes de professores, cada um com um checkbox para seleção:

- Edmilson Campos
- Sandra Camara
- Adriano Costa
- Alba Lopes
- Ailton Camara
- Alex Wagner
- Alinne Pompeu
- Aparecida Rego

Fonte: Equipe ACHA

Figura 22 - Editar área

Artes

[Início](#) / [Áreas](#) / [Artes](#)

ID: 18

Nome: Artes

Subárea:

Grupo: Ciências Humanas e Sociais

Submeter VOLTAR

Fonte: Equipe ACHA

Figura 23 - Editar disciplina

Arquitetura de Sistemas

[Início](#) / [Disciplinas](#) / [Arquitetura de Sistemas](#)

ID: 68

Nome: Arquitetura de Sistemas

Carga Horária: 60

Área: Eletrônica

Submeter VOLTAR

Fonte: Equipe ACHA

4.2 SUMÁRIO

Observou-se que nesse capítulo foram abordados os aspectos do SAD ACHA em si, falando de suas características e usabilidade, com foco no entendimento do sistema como um todo. Por fim, o próximo capítulo apresenta as considerações resultantes da produção deste documento.

5 CONSIDERAÇÕES FINAIS

Diante do que foi apresentado podemos ver que a modernização dos sistemas de informações ajudam a agilizar e fazer com que tudo seja mais simples para os usuários, facilitando a vida daqueles que os utilizam. Com o ACHA percebemos que o aperfeiçoamento do padrão de montar as matrizes dos cursos pode minimizar o trabalho de muitos servidores, evitando que ocorram erros, e caso existam, sejam mais facilmente solucionados, fazendo com que essa tarefa deixasse de ser algo demorado e cansativo e passasse a ser algo mais fácil e rápido reduzindo o número de pessoas envolvidas.

Além disso, o desenvolvimento do projeto permitiu a fixação dos conhecimentos adquiridos no decorrer do curso pelos alunos, assim como a possibilidade de compor o trabalho de conclusão de curso para os envolvidos. Somando-se a isso, também existe o fator de contribuir com a comunidade do IFRN utilizando o que foi aprendido no decorrer de nossa convivência com esta.

Entretanto, o sistema, atualmente, não pode ser considerado completo, uma vez que, da lista de 19 requisitos funcionais coletados durante o processo de desenvolvimento, os seguintes não foram cumpridos: gerenciamento de campus, exportação de dados, importação de dados, gerenciamento de planejamentos, definição de coordenadores de cursos, distribuição de professores.

Mesmo assim, existe, para o grupo, um sentimento de dever cumprido com a entrega do projeto, além de esperança que, mais a frente, o mesmo possa ser aprimorado ainda mais.

5.1 ATIVIDADES FUTURAS

Com a entrega do resultado deste primeiro ciclo de desenvolvimento do projeto, espera-se que, através do *feedback* apresentado pelos usuários do sistema, novos contribuintes para o projeto possam melhorar o sistema a fim de adequá-lo ainda mais as necessidades dos operadores.

Além disso, a conclusão de requisitos não entregues até esta fase do projeto devem ser prioridade para os próximos responsáveis, uma vez que são um incremento já previsto para projeto.

REFERÊNCIAS

BARBOSA, Geane. **Sistema de Apoio à Decisão SAD**. 2008. Disponível em: <<http://www.administradores.com.br/artigos/tecnologia/sistema-de-apoio-a-decisao-sad/26378/>>. Acesso em: 1 nov. 2016.

BOOCH, Grady; RUMBAUGH, James; JACOBSEN, Ivar. **The Unified Modeling Language User Guide**. 2. ed. Reading: Addison-wesley, 2005.

CLEBER, Christian. **Introdução ao Processo Unificado**. Disponível em: <<http://www.devmedia.com.br/introducao-ao-processo-unificado/3931>>. Acesso em: 1 nov. 2016.

CSS. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/CSS>>. Acesso em: 03 out. 2016.

DALMÁS, Ângelo. **Planejamento participativo na escola**. 14. ed. Petrópolis: Vozes, 2008.

EDUARDO, Marcio. **O que é Planejamento?** 2008. Disponível em: <<http://www.administradores.com.br/artigos/negocios/o-que-e-planejamento/39381/>>. Acesso em: 1 nov. 2016.

ELMASRI, Ramez; NAVATHE, Shamkant B.. **SISTEMAS DE BANCO DE DADOS**. 4. ed. São Paulo: Pearson Education, 2005.

HTML. 2016. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTML>>. Acesso em: 24 out. 2016.

ILVA, Giancarlo. **Gerencie equipes e tarefas com o Trello e dê adeus aos post-its!**. Disponível em: <<https://corporate.canaltech.com.br/dica/utilitarios/gerencie-equipes-e-tarefas-com-o-trello-e-de-adeus-aos-post-its/>>. Acesso em: 03 out. 2016.

JAQUES, Rafael. **O que é um Framework? Para que serve?** 2016. Disponível em: <<http://www.phpit.com.br/artigos/o-que-e-um-framework.phpit>>. Acesso em: 1 nov. 2016.

JAVASCRIPT. 2016. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acesso em: 03 out. 2016.

LEITE, Jair C. **Planejamento e Gerenciamento de Projetos de Software.** 2007. Disponível em: <<http://engenhariadesoftware.blogspot.com.br/2007/04/planejamento-e-gerenciamento-de.html>>. Acesso em: 1 nov. 2016.

MBCONTROL. **Processo automático x Processo Manual:** A vantagem em utilizar um software para fazer a Gestão de Telecom dentro da sua empresa. 2014. Disponível em: <http://mobilecontrol.com.br/blog/processo-automatico-x-processo-manual-a-vantagem-em-utilizar-um-software-para-fazer-a-gestao-de-telecom-dentro-da-sua-empresa/#.WA4BFd_6z0o>. Acesso em: 24 out. 2016.

PORTOGENTE. **Sistema de Apoio à Decisão.** 2016. Disponível em: <<https://www.portogente.com.br/portopedia/74775-sistema-de-apoio-a-decisao>>. Acesso em: 1 nov. 2016.

PRESSMAN, Roger S.. **Engenharia de Software:** Uma Abordagem Profissional. 7. ed. New York: The McGraw-hill, 2011. 780 p.

PRIBERAM. **Brainstorming.** Disponível em: <<https://www.priberam.pt/dlpo/brainstorming>>. Acesso em: 24 out. 2016.

RAMOS, Allan. **MVC – Afinal, é o quê ?** 2015. Disponível em: <<http://tableless.com.br/mvc-afinal-e-o-que/>>. Acesso em: 24 out. 2016.

RODRIGUES, Joel. **Modelo Entidade Relacionamento (MER) e Diagrama Entidade-Relacionamento (DER).** Disponível em:

<<http://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332>>. Acesso em: 24 out. 2016.

SCHACH, Stephen R.. **Engenharia de Software**: Os paradigmas clássicos orientado a objetos. 7. ed. São Paulo: Mcgraw-hill, 2009.

SCHMITZ, Daniel. **Tudo que você queria saber sobre Git e GitHub, mas tinha vergonha de perguntar**. 2015. Disponível em:<<http://tableless.com.br/tudo-que-voce-queria-saber-sobre-git-e-github-mas-tinha-vergonha-de-perguntar/>>. Acesso em: 03 out. 2016.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson Education, 2011.

VIEIRA, Denisson. Scrum: **A Metodologia Ágil Explicada de forma Definitiva**. 2014. Disponível em: <<http://www.mindmaster.com.br/scrum/>>. Acesso em: 03 out. 2016.

ANEXO I - GLOSSÁRIO

ÁREA: Campo em que os professores exercem suas atividades, onde cada um é levado a uma área baseado no seu tipo de conhecimento.

CARGA HORÁRIA: quantidade de créditos que um componente curricular consome semanalmente.

COMPONENTE CURRICULAR: subdivisão da disciplina que ocorre em um período. Apresenta o número de efeito para cálculos (carga horária).

Tabela 3 - Exemplo componentes curriculares

ex:	Período 1	Período 2	Período 3
Química	x	Química I	Química II
Geografia	Geografia I	Geografia II	x
Matemática	Matemática I	Matemática II	Matemática III

Legenda: **Disciplinas;** **Componente curricular**

Fonte: Equipe ACHA

CRÉDITO: Número de aulas semanais que cada professor pode dar, por exemplo, 2 créditos são 2 aulas numa semana para determinada turma.

CURSO: Conjunto de disciplinas que normalmente são voltadas para a preparação profissional onde ao fim se recebe um diploma.

DIRETORIA ACADÊMICA: Setor responsável por dirigir os processos acadêmicos no campus.

DISCIPLINA: É um conjunto de componentes curriculares que compõem a grade curricular de uma matriz de um curso de uma determinada modalidade.

FASE BETA: é a fase de um software em que os testes são efetuados e que ele começa a ganhar sua forma.

GRUPO DE PROFESSORES: São os os professores que se responsabilizam por determinada área do conhecimento acadêmico definido um coordenador, normalmente numa área técnica, e selecionando as disciplina que serão lecionadas.

HORA/AULA: Tempo, normalmente de 45 a 50 minutos, em que o professor está lecionando.

HORA/RELÓGIO: Diz respeito à diferença entre a hora-aula de 50 minutos praticada pela Unidade de Ensino e a hora-aula de 60 minutos dita relógio.

MATRIZ: grade curricular de um curso; conjunto de disciplinas e suas respectivas cargas horárias de um curso.

MODALIDADE: tipo de curso. ex: integrado, subsequente, licenciatura.

NÚCLEO: núcleo politécnico correspondente ex: estruturante, articulador, tecnológico.

OFERTAR: Processo de abrir novas turmas realizadas ao fim de cada período.

OFERTA: um dos resultados do planejamento, compreende às turmas dos determinados cursos que ingressarão nos próximos semestres.

PERÍODO: Tempo no qual um curso se renova, normalmente relativo a abertura de novas turmas. Nos cursos superiores e técnicos é um semestre e nos cursos técnico integrados um ano.

PLANEJAMENTO: é a relação entre as turmas, disciplinas e a carga horária, de modo que uma turma tem diversas disciplinas e essas por sua vez somam a carga horária que será ocupada pelos professores.

PLANILHA: é uma tabela eletrônica utilizada na maioria das vezes para organizar dados.

PROTÓTIPOS: equivalem ao sistema ainda não pronto, ou seja quando está em sua fase beta ou posterior, desde de que não esteja completo.

STAKEHOLDER: principais pessoas interessadas no sistema.

SUBÁREA: delimitação com maior grau de especificidade da área.

VAGA: uma vaga é aberta em uma turma em determinado semestre para uma disciplina de modo que o professor possa assumi-la.

VISUALIZAR VIABILIDADE: quando o usuário simular a carga horária, será emitido um panorama com as informações sobre conflitos entre créditos adicionados e o atual corpo docente. Dessa forma, o diretor acadêmico terá consciência da necessidade de contratação (ou não) dos professores.