

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Diego de Pontes Pasquini

**Implementação de Sistema Web para
Distribuição Didática da FAMAT-UFU
utilizando método SCRUM e pilha MEAN**

Uberlândia, Brasil

2019

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Diego de Pontes Pasquini

**Implementação de Sistema Web para Distribuição
Didática da FAMAT-UFU utilizando método SCRUM e
pilha MEAN**

Trabalho de conclusão de curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, Minas Gerais, como
requisito exigido parcial à obtenção do grau
de Bacharel em Ciência da Computação.

Orientador: Profa. Maria Adriana Vidigal de Lima

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2019

Diego de Pontes Pasquini

Implementação de Sistema Web para Distribuição Didática da FAMAT-UFU utilizando método SCRUM e pilha MEAN

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Ciência da Computação.

Trabalho aprovado. Uberlândia, Brasil, XX de junho de 2019.

Profa. Maria Adriana Vidigal de Lima
Orientadora

Convidado 1

Convidado 2

Uberlândia, Brasil
2019

Resumo

Este trabalho aborda os aspectos teóricos e práticos necessários para o desenvolvimento de um sistema de Gerenciamento e Distribuição Didática para docentes da Faculdade de Matemática na Universidade Federal de Uberlândia. Além dos requisitos específicos relativos à distribuição de disciplinas, este trabalho aborda aspectos de engenharia de software que são necessários para a eficácia na implementação. Este trabalho apresenta também discussões sobre arquitetura, metodologias, funcionalidades e outros aspectos relevantes para a criação de um software com qualidade.

Palavras-chave: engenharia de software, metodologias, requisitos, desenvolvimento web.

Lista de ilustrações

Figura 1 – Arquitetura Cliente-Servidor.	13
Figura 2 – Métodos HTTP Utilizados na Comunicação Cliente-Servidor.	13
Figura 3 – Arquitetura de uma aplicação web utilizando as tecnologias MEAN. . .	15
Figura 4 – Fluxo da Divisão do Corpo Docente Em Grupos.	18
Figura 5 – Arquitetura do sistema.	24
Figura 6 – Interação entre os elementos do sistema para o caso de uso de um fechamento de período de registro de atividades.	25
Figura 7 – Objetos no Banco de Dados.	26
Figura 8 – Organização dos diretórios de código da aplicação.	28
Figura 9 – Visão Geral do Processo de Distribuição de Disciplinas.	40
Figura 10 – Fluxo do Registro de Atividades dos Docentes.	41
Figura 11 – Fluxo do Fechamento do Período de Registro de Atividades: Atribuição de pontos	44
Figura 12 – Fluxo do Fechamento do Período de Registro de Atividades: Atribuição de grupos	45
Figura 13 – Fluxo da Escolha de Turmas.	49
Figura 14 – Fluxo do Fechamento do Período de Seleção de Turmas: Distribuição. .	50
Figura 15 – Pior Caso De Distribuição.	51

Lista de abreviaturas e siglas

API	<i>Application Programming Interface</i>
HTML	<i>Hyper Text Markup Language</i>
CSS	<i>Cascading Style Sheets</i>
JS	<i>JavaScript</i>
UML	<i>Unified Modeling Language</i>
MVC	<i>Model View Controller</i>
FAMAT	<i>Faculdade de Matemática</i>
IO	<i>Input/Output</i>
XML	<i>eXtensible Markup Language</i>
JSON	<i>JavaScript Object Notation</i>
REST	<i>Representational State Transfer</i>
URL	<i>Uniform Resource Locator</i>
URI	<i>Uniform Resource Identifier</i>

Sumário

1	INTRODUÇÃO	8
1.1	Visão Geral	8
1.2	Justificativas	8
1.3	Objetivos	9
1.4	Organização da Monografia	9
2	REFERENCIAL TEÓRICO	10
2.1	Engenharia de Software	10
2.1.1	Processo de Software	10
2.1.2	Método Scrum	11
2.2	Desenvolvimento Web de Três Camadas	12
2.3	Padrão de Arquitetura REST	12
2.4	Banco de Dados NoSQL	14
2.5	MEAN	15
3	SISTEMA DE DISTRIBUIÇÃO DE DISCIPLINAS FAMAT	17
3.1	Procedimento de Distribuição de Disciplinas	17
3.2	Requisitos Funcionais e Não Funcionais	18
3.2.1	Prioridade dos Requisitos	18
3.2.2	Requisitos Funcionais	19
3.2.3	Requisitos Não Funcionais	22
3.3	Arquitetura do Sistema	24
3.4	Projeto de Banco de Dados Orientado a Documento	26
3.5	Desenvolvimento SCRUM	27
3.5.1	Backlog do Produto	27
3.5.2	Sprint 1	28
3.5.2.1	Planejamento	28
3.5.2.2	Execução	28
3.5.2.3	Revisão do Sprint	40
3.5.3	Sprint 2	40
3.5.3.1	Backlog	40
3.5.3.2	Execução	40
3.5.3.3	Revisão do Sprint	46
3.5.4	Sprint 3	46
3.5.4.1	Backlog	46
3.5.4.2	Execução	46

3.5.4.3	Revisão do Sprint	52
4	CONCLUSÃO E TRABALHOS FUTUROS	53
	REFERÊNCIAS	54
	ANEXO A – CICLO DE DISTRIBUIÇÃO DE DISCIPLINA - RE- SOLUÇÃO 02/2017 - FAMAT	55

1 Introdução

Este trabalho descreve o desenvolvimento de um sistema de gerenciamento, como ferramenta para auxiliar o processo de gestão da distribuição didática da FAMAT, desde a validação de atividades dos docentes, até a distribuição das disciplinas entre os professores. Utilizando a ferramenta, os docentes poderão relatar suas atividades acadêmicas, bem como verificar o estado de aprovação de cada uma delas. Além disso, cada docente poderá ver seu desempenho e receber feedbacks relacionados às suas atividades. A coordenação poderá avaliar as atividades dos docentes, fazer ajustes necessários a cada disciplina e atividade, bem como utilizar o sistema para facilitar a própria distribuição da carga de disciplinas de cada docente dentro da FAMAT.

1.1 Visão Geral

Atualmente os docentes da FAMAT precisam preencher um extenso formulário todos os anos, para que a comissão possa avaliar e atribuir uma pontuação particular a cada docente. Este formulário envolve a verificação das atividades de Ensino, Pesquisa e Extensão. Após a contagem dos pontos, os professores são classificados, em ordem decrescente respeitando a pontuação obtida. Esta classificação contém três grupos, o grupo G3 (ministrando no mínimo 8ha), o grupo G2 (ministrando no mínimo 10ha) e o grupo G1 (ministrando no mínimo 12ha). É feita então, de acordo com uma lista de preferência particular de cada professor, a distribuição de disciplinas. Assim sendo, a comissão realiza manualmente por um longo período de tempo, vários procedimentos que poderiam facilmente ser realizados por um software.

1.2 Justificativas

Devido a uma proposta de distribuição de disciplinas que torne as cargas de trabalho de cada docente igualitária e justa, os professores da Faculdade de Matemática da Universidade Federal de Uberlândia se deparam a cada semestre com uma distribuição de disciplinas, cuja atividade é desgastante e burocrática. Esta situação impõe aos docentes, a dedicação de uma quantidade de tempo relevante, pois o processo é feito sem o auxílio efetivo de qualquer tecnologia.

1.3 Objetivos

O objetivo deste trabalho é aplicar o conhecimento adquirido nas disciplinas cursadas ao longo do curso de Bacharelado em Ciências da Computação na criação de uma aplicação web para gerenciar a distribuição de disciplinas entre os docentes da FAMAT, digitalizando o processo de escolha de disciplinas e a declaração do trabalho realizado no ano e, conseqüentemente automatizando o processo. Espera-se que a criação deste sistema e a automatização de todo o processo, possa contribuir para a maior facilidade, agilidade e eficácia na distribuição de disciplinas.

No sistema, cada professor poderá escolher as disciplinas que tiver interesse, poderá declarar seus trabalhos realizados e a sua carga horária, sem a necessidade da participação da coordenação nesta atividade. E os membros da comissão poderão realizar a distribuição em um tempo curto e sem a necessidade de fazer manualmente todas as validações e atribuições das turmas. A automação visa evitar situações de retrabalho por qualquer erro humano.

Este trabalho propõe o desenvolvimento do software supracitado utilizando-se a arquitetura MEAN, que envolve uma coleção de tecnologias baseadas em JavaScript utilizadas para desenvolver aplicações web, e uma metodologia de desenvolvimento ágil. Desta forma, pretende-se a construção de uma aplicação de software com código de qualidade e de fácil expansão para a adição de novas funcionalidades futuramente.

1.4 Organização da Monografia

Este documento está organizado da seguinte forma: no primeiro capítulo apresenta-se o projeto com uma breve introdução ao tema e do problema proposto, bem como os objetivos gerais e específicos. O segundo capítulo descreve todas as tecnologias e conceitos a serem utilizados durante o desenvolvimento do projeto. No terceiro capítulo são apresentadas as etapas realizadas para a confecção do software, e como o projeto foi desenhado e implementado, apontando os requisitos coletados, a estrutura estabelecida inicialmente e a estrutura da versão final do projeto, além da descrição de todo o processo de software. As conclusões deste trabalho estão registradas no quarto capítulo.

2 Referencial Teórico

Neste capítulo serão apresentados os principais conceitos utilizados no desenvolvimento do sistema, bem como na análise de requisitos. Também serão explicadas algumas definições para o entendimento da modelagem do sistema.

2.1 Engenharia de Software

Com o objetivo de aprimorar o desenvolvimento de software em sua produtividade e na qualidade do produto final, surge a Engenharia de Software como sendo uma área da Ciência da Computação voltada à especificação, desenvolvimento e manutenção de sistemas de software. Esta área trata de aspectos relacionados ao estabelecimento de processos, métodos, técnicas, ferramentas e ambientes de suporte ao desenvolvimento de software (FALBO, 2005). A importância de tal disciplina é evidenciada quando percebe-se a dependência da sociedade e seus indivíduos, em softwares mais complexos. Pois assim, exige a capacidade de uma produção de softwares confiáveis (SOMMERVILLE, 2011). Com o desenvolvimento da engenharia de software, surgiram também os diagramas comportamentais e estruturais. Dos quais o primeiro é a representação visual das funções que deverão ser implementadas e as ações esperadas pelo cliente e o último é a representação da arquitetura/estrutura do software em diferentes escopos. São alguns deles:

Diagrama de caso de uso Informa as ações esperadas da aplicação em um caso específico de requisição do usuário, sequencialmente e de uma forma abstrata sem conteúdo técnico.

Diagrama de implantação Informa como o software deve ser implantado, com a descrição do ambiente de hardware ou software para que seja executado.

2.1.1 Processo de Software

Um processo de software pode ser definido como um conjunto de atividades relacionadas, que levam à produção de um software. Ou seja, é uma abordagem sistemática da engenharia de software que possui quatro atividades comuns:

Especificação de software Onde ocorre a definição do software idealizado e as restrições de sua operação.

Desenvolvimento de software Onde ocorre a tradução do projeto de software para linguagem de programação.

Validação de software Pela qual o software é avaliado para constatar se atende as necessidades do cliente.

Evolução de software Onde o produto é modificado para refletir as mudanças de requisitos do cliente e do mercado.

Assim sendo, existem representações simplificadas e adaptadas do processo em vários modelos genéricos, cujos modelos são abstrações usadas para diferentes abordagens no desenvolvimento de software. Um dos principais modelos é o Modelo em Cascata, que utiliza cada etapa do processo de software como fases distintas e de forma sequencial. Neste modelo, cada estágio não pode ser iniciado até que o anterior esteja concluído com documentos (“assinados”). Assim sendo, é comum que se congele algumas etapas do processo, podendo gerar um sistema diferente do esperado pelo cliente, uma vez que as soluções são deixadas para uma etapa posterior. E quando está no estágio final de testes, podem ser descobertos erros e necessidades funcionais que implicariam em repetições de etapas anteriores. Isto torna o modelo em cascata próprio para projetos que necessitam de uma especificação detalhada e certa na sua definição, como sistemas críticos. Já para os projetos que podem ter seus requisitos alterados, desenvolveram-se novos modelos, a saber, primeiramente o desenvolvimento incremental e posteriormente as metodologias ágeis (SOMMERVILLE, 2011).

2.1.2 Método Scrum

O método Scrum é classificado como um método ágil, e tem como objetivo atender a necessidade de desenvolvimento rápido e lidar com requisitos mutáveis. Isto já era uma necessidade quando o modelo de desenvolvimento incremental foi sugerido e, por isso, o Scrum compartilha com tal método muitas características: existência de uma especificação parcial e intercalada com a implementação e desenvolvimento em uma série de versões. Para os métodos ágeis, os incrementos são menores e possuem versões disponibilizadas para o cliente para obter uma avaliação rápida sobre a evolução dos requisitos (SOMMERVILLE, 2011).

Pode-se dividir todo o processo de desenvolvimento com Scrum em três etapas. A primeira sendo de planejamento, na qual faz-se uma análise do produto de forma geral e determina-se a arquitetura do software e dos recursos a serem utilizados. A segunda sendo de desenvolvimento é composta de vários ciclos ou *sprints* - no início de cada ciclo é realizado o *backlog* do produto, onde faz-se uma lista do que vai ser realizado e são verificados os riscos e as prioridades. E por fim, a terceira etapa sendo a qual se finalizam as documentações ou manuais, e entrega-se do software ao *stakeholder* ou cliente (DIMES; BARROS, 2014).

2.2 Desenvolvimento Web de Três Camadas

A maioria das aplicações para internet são feitas na arquitetura de três camadas que consiste em três divisões importantes: dados, lógica, e apresentação. Nas aplicações web, a estrutura usualmente se resume a banco de dados, servidor, e cliente, enquanto no desenvolvimento moderno de aplicações web, também pode-se optar por definir camadas para banco de dados, lógica de servidor, lógica de cliente e interface do usuário. Um paradigma popular na implementação deste modelo é o padrão MVC (*Model View Controller*). Este último separa a lógica, os dados e a visualização em três partes, cada uma com sua tarefa única. A *View* (“visão”) controla a parte visual e os mecanismos de interação com o usuário. O *Controller* (“controlador”) responde à eventos do sistema e do usuário, coordenando as camadas *Model* e *View* para que mudem apropriadamente. A *Model* (“modelo”) possui a tarefa de manipulação de dados, respondendo às requisições de informações ou de alterações de acordo com as instruções do Controller (HAVIV, 2016). É importante notar também que a estrutura em camadas das aplicações web pode ser agrupada em duas categorias gerais:

Front-end: Sendo composto pela camada View do padrão MVC. Possui as funções voltadas para a interação direta com o usuário.

Back-end: Sendo composto pelas camadas Model e Controller, possui a função de armazenar e processar dados para atender às requisições do front-end que é executado no lado do cliente/usuário.

2.3 Padrão de Arquitetura REST

Representational State Transfer traduzido como Transferência Representacional de Estado, propõe que aplicações web deveriam seguir alguns padrões arquiteturais específicos dentre os que já existem e ainda define um conceito de disponibilização de recursos. Arquiteturalmente uma aplicação RESTful deve ser composta de duas partes distintas, cliente e servidor.

Dentre as duas partes há então uma distinção de responsabilidades, deixando que as funções de controle de sessões e geração de interface sejam feitas pelo lado do cliente e as funções de armazenamento e manipulação de dados sejam feitas pelo lado do servidor. Com esta divisão temos que a arquitetura cliente-servidor passa a ser *stateless*, significando que em ambas as partes não existe qualquer registro de estado. Portanto, as requisições enviadas ao servidor precisam conter todas as informações necessárias para que sejam compreendidas e processadas. O lado negativo desse processo seria a redução de desempenho em rede devido ao aumento na repetição dos dados nas requisições.

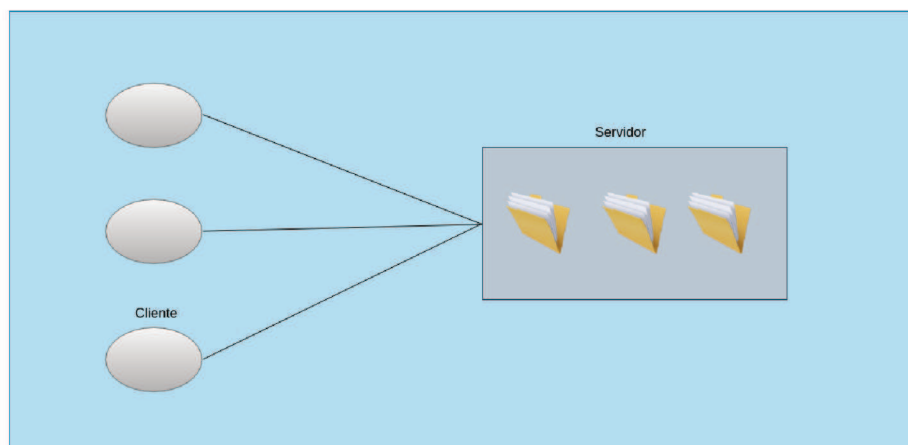


Figura 1 – Arquitetura Cliente-Servidor.

A comunicação entre as duas partes é realizada através das operações do protocolo HTTP. Os métodos mais comuns são:

GET Usado principalmente para a busca de dados no servidor

POST Método aplicado majoritariamente para a inserção de informações no servidor

DELETE Como um meio para a deleção de dados.

Nos servidores web arquitetados de forma RESTful todos os recursos (pontos de acesso) são identificados através da URI e a representação dos dados a serem transferidos pode feita através dos formatos: JSON, XML ou HTML. Na Figura 2 está ilustrada a utilização dos métodos mencionados e dos recursos disponibilizados pelo servidor pelo lado cliente.

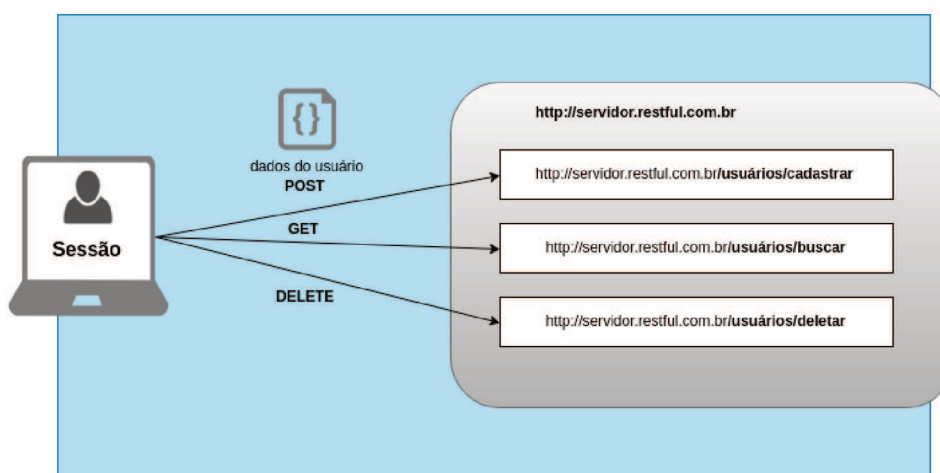


Figura 2 – Métodos HTTP Utilizados na Comunicação Cliente-Servidor.

2.4 Banco de Dados NoSQL

Com o surgimento da internet e computação em nuvem, as aplicações mais recentes fizeram com que surgissem novas necessidades com relação aos bancos de dados (HAN et al., 2011). São elas:

Alta concorrência de leitura e escrita com baixa latência: existia uma demanda por bancos de dados que pudessem responder rapidamente mesmo que houvesse uma alta concorrência de *IO*. Tudo para que não houvesse gargalo (um processo bloqueando ou atrasando a continuidade de outros processos) e a aplicação pudesse reagir mais rapidamente.

Alta escalabilidade e disponibilidade: em busca de um serviço sem interrupções, havia a necessidade de uma base de dados que pudesse ser expandida e aprimorada conforme o aumento do número de requisições concorrentes.

Eficiência em grandes quantidades de dados e acessos: à medida em que número de dados e requisições aumenta, maiores são as dificuldades em se utilizar uma base relacional. Portanto, para grandes aplicações era necessário que existissem bases de dados que lidassem com milhões de requisições e uma vasta quantidade de informações eficientemente.

Para atender às necessidades acima surgiram novas bases, que em sua maioria são diferentes dos bancos relacionais tradicionais. (HAN et al., 2011). Tais bancos são classificados em diferentes tipos, de acordo com a sua estrutura. São eles:

Chave-valor: São os bancos que possuem uma estrutura composta de uma chave e um valor relacionado a ela. Mesmo que sua estrutura seja tão simples, esta base suporta uma enorme quantidade de dados e alta concorrência, com uma velocidade de resposta maior que a dos tradicionais bancos relacionais.

Orientado a colunas: Os dados são armazenados por colunas, sendo que cada coluna de dados é o índice utilizado no banco. Assim, reduz-se a quantidade de operações de leitura e escrita do sistema, já que cada requisição acessa somente a coluna que contém a informação desejada. A escolha desta estrutura é vantajosa para aplicações de data-warehouse.

Orientado a Documentos: São similares aos de chave-valor em sua estrutura, já que a diferença está no valor, que é semântico (geralmente em formato XML ou JSON). E diferencia-se também por permitir que um valor contenha referência a um outro índice para que torne mais fácil para as aplicações o tratamento de referências.

Orientado a Grafos: Composto de nós e arestas, onde os nós se relacionam por meio de arestas. Assim, os nós são pares de chave/valor e os relacionamentos têm sempre um nome associado e uma direção contendo um nó remetente e outro destinatário, podendo conter também propriedades. Com esta estrutura e um número ilimitado de relações, permite-se então que a interpretação dos dados seja diferente com base nas diversas relações (CUNHA; PEREIRA, 2016).

Tais bancos são categorizados como NoSQL, traduzido como “Não SQL”, significando que não são bancos onde os dados são modelados por meio de relações de tabelas como os bancos tradicionais.

2.5 MEAN

Nesta seção serão expostas as tecnologias que foram escolhidas para o desenvolvimento do Sistema de Gerenciamento de Disciplinas da FAMAT. Tais tecnologias juntas, formam o conjunto MEAN de desenvolvimento web, que foi escolhido por possuir alta performance, facilidade de desenvolvimento, facilidade de manutenção e facilidade de manutenção da aplicação (HAVIV, 2016) .

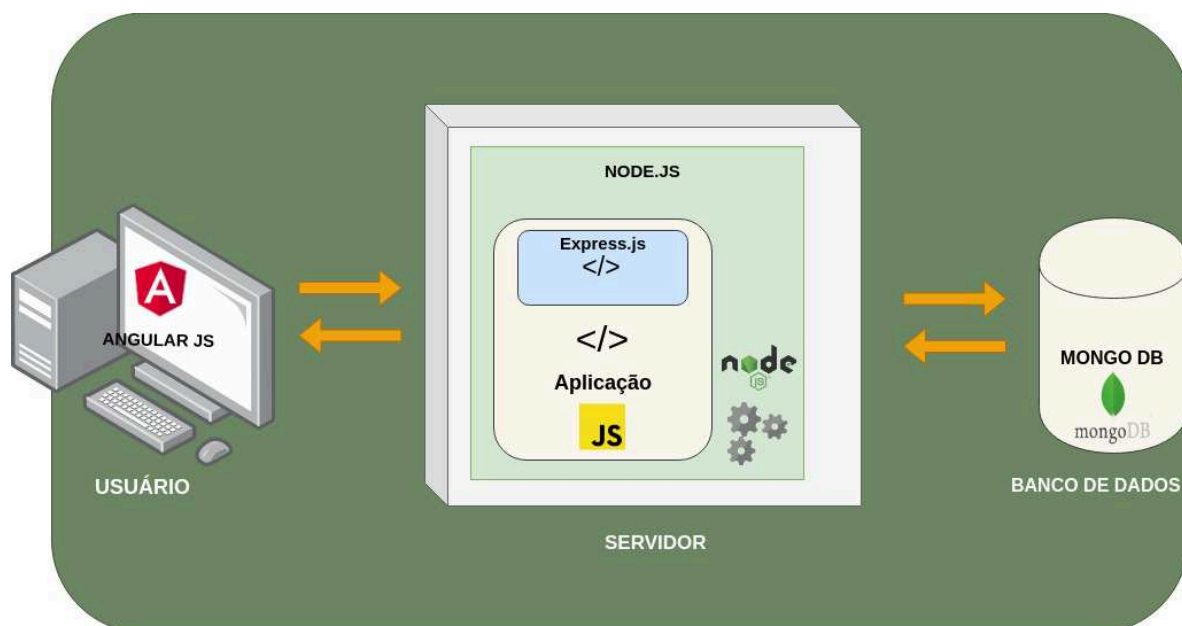


Figura 3 – Arquitetura de uma aplicação web utilizando as tecnologias MEAN.

MEAN é uma abreviação para as tecnologias das quais é composta, sendo todas elas *OpenSource* e padronizadas em uma linguagem (JavaScript), o que é comumente chamado de *full-stack* JavaScript. São elas:

MongoDB: MongoDB é um banco de dados NoSQL orientado a documentos, que utiliza como valor dos dados a semântica JSON (*JavaScript Object Notation*) que é o

formato usado nos objetos do JavaScript. Assim os dados podem ser facilmente armazenados e obtidos sem que haja serialização ou desserialização de informações durante as operações entre o banco e as aplicações ([CHODOROW, 2013](#)).

ExpressJS: É um framework para criação de servidores web em JavaScript em conjunto com o NodeJS. Disponibiliza portanto funções necessárias para facilitar e tornar mais ágil o desenvolvimento e a configuração de APIs para servidores em aplicações web.

Angular: É um framework que tem como princípio criar uma ponte entre o JavaScript e o HTML/CSS, e incentivar a prática de desenvolvimento web de uma página. Sendo arquiteturalmente padronizada em camadas (MVC), o Angular permite que haja uma sincronização de dados entre a camada de modelo e a camada de visão, facilitando a manipulação de dados entre as mesmas ([HAVIV, 2016](#)).

NodeJS: Apesar do JavaScript originalmente ter sido apenas utilizado para programação voltada para navegadores, o seu grande potencial enquanto linguagem acarretou no interesse em utilizá-lo no desenvolvimento de outros tipos de aplicações. O NodeJS, que é um interpretador de JavaScript, nasce então para a execução de aplicações JS em servidores. Apesar de não ser o primeiro, tem o diferencial de ter uma arquitetura orientada a eventos, que permite processar requisições de forma assíncrona. O que é extremamente vantajoso para um servidor, que passa a ter maior facilidade em receber um número grande de requisições e responde-las em um tempo hábil.

3 Sistema de Distribuição de Disciplinas FAMAT

3.1 Procedimento de Distribuição de Disciplinas

Nesta seção serão descritos os pontos fundamentais do procedimento que é realizado pela FAMAT para a distribuir as disciplinas aos professores, e que serão automatizados e incorporados no sistema. Cada professor em um período de tempo determinado pode declarar suas atividades realizadas no semestre. Cada atividade deve ser enquadrada em pelo menos um dos 4 anexos de atividades predeterminados pela Comissão de Distribuição de Aulas. Com as atividades submetidas e as pontuações atribuídas, todos os docentes efetivos ativos serão divididos em três grupos: G3, G2 e G1. Sendo o primeiro composto por docentes melhor colocados que declararam pelo menos uma atividade em cada um dos anexos 1, 2 e 3. Já os usuários que declararem uma atividade em dois dos três anexos 1,2 e 3 poderão pertencer ao grupo G2, bem como ao grupo G1, dependendo da pontuação verificada.

Dentro de cada grupo, os docentes serão ordenados de acordo com a pontuação recebida pelas atividades submetidas. Todos os docentes possuem o limite máximo de 16 horas por semana, porém cada grupo dá ao professor uma carga horária semanal mínima diferente. Sendo 8, 10 e 12 as horas aula semanais dos grupos G3,G2 e G1 respectivamente. Porém caso o professor tenha papéis administrativos (de coordenação, por exemplo) a carga semanal mínima deve ser reduzida para 6 h/a. Caso o docente seja assessor administrativo sua menor carga semanal será de 8 horas. Cada grupo exige uma escolha de disciplinas diferente para o professor: os docentes do grupo G3 precisam escolher disciplinas que possuem uma carga horária semanal maior que quatro horas aula, os docentes do grupo G2 precisam escolher disciplinas que possuem uma carga horária semanal maior que cinco horas aula e os docentes do grupo G1 precisam escolher disciplinas que possuem uma carga horária semanal maior que seis horas aula.

Durante um período determinado pela comissão de Distribuição de Aulas os docentes poderão escolher pelo menos quinze disciplinas que têm maior interesse em lecionar. Aqueles que estão no grupo G3 poderão escolher apenas as disciplinas com carga horária semanal maior ou igual 4 horas ou disciplinas do grupo SM/SE. Os que estão no grupo G2 poderão escolher apenas aquelas disciplinas que têm no mínimo 5 horas semanais de aula ou disciplinas do grupo SM/SE, e os do G1 as com carga maior ou igual a 6 horas ou disciplinas do grupo SM/SE. Exceto os professores com atividades administrativas ou assessores administrativos, que não têm tais restrições.

Com a lista de disciplinas escolhidas, devidamente ordenadas pela preferência do docente, a comissão pode então escolher todos os três grupos ou parte deles para atribuir as turmas. Para realizar a distribuição considera-se uma ordem de preferência entre os três grupos, na qual o grupo G3 está no topo com o grupo G2 em seguida e por fim o G1. Esta organização define a ordem pela qual a distribuição acontece, atribuindo primeiro uma turma do grupo de maior preferência. Assim uma distribuição será composta de vários turnos, nos quais atribui-se apenas uma turma por vez aos docentes. Quando o último docente recebe uma turma, um novo turno começa novamente pelo primeiro professor da lista. Poderão ser realizados quantas distribuições a comissão achar necessário.

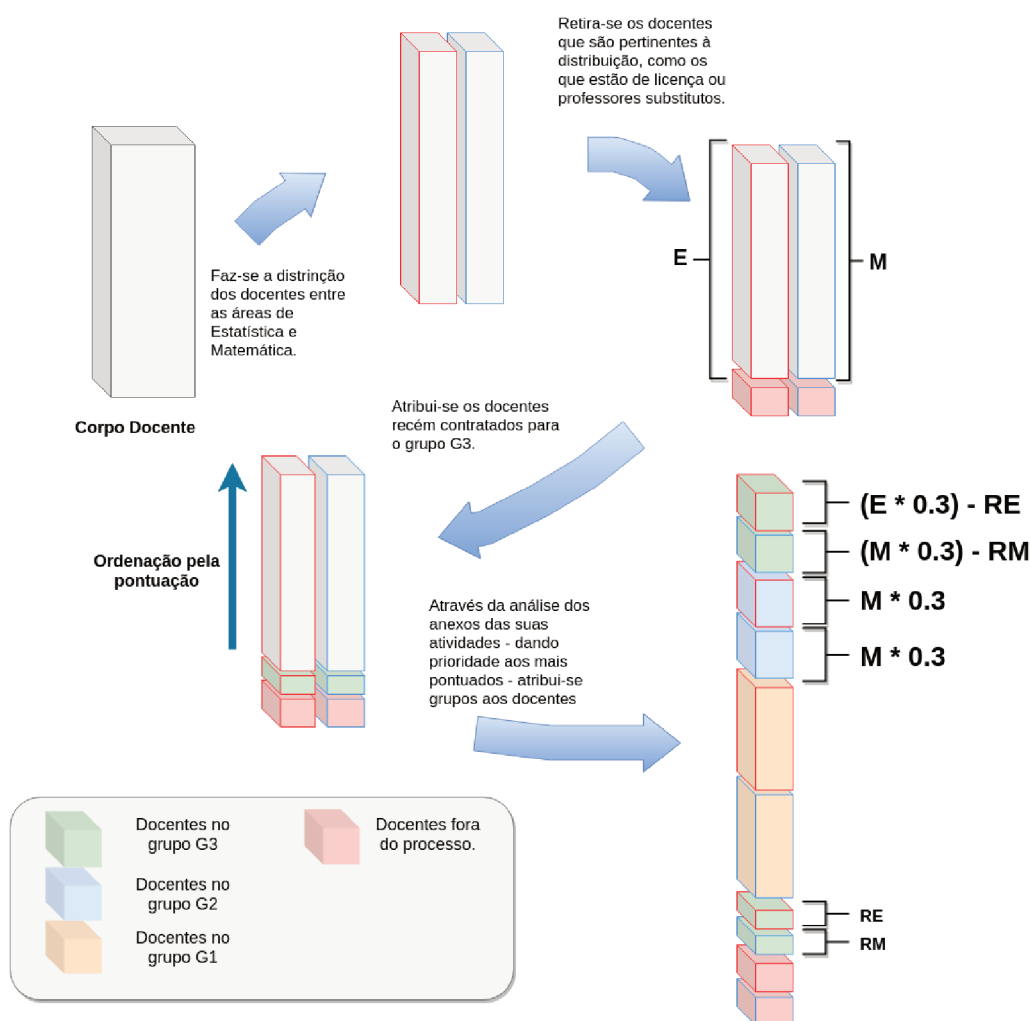


Figura 4 – Fluxo da Divisão do Corpo Docente Em Grupos.

3.2 Requisitos Funcionais e Não Funcionais

3.2.1 Prioridade dos Requisitos

Os requisitos funcionais descrevem a funcionalidade ou os serviços que o sistema deve realizar em benefício dos usuários (FILHO, 2003; BRAUDE; BERNSTEIN, 2016).

Estes requisitos são específicos de cada software, considerando seu tipo e os seus usuários e podem ser expressos de diversas maneiras e em diferentes níveis de detalhamento. Normalmente o conjunto de requisitos de um software é bastante extenso, e é importante definir a prioridade dos mesmos para que se determine quais deles devem ser implementados em primeiro lugar. Para estabelecer a prioridade dos requisitos foi adotado o seguinte padrão:

Essencial: É o requisito sem o qual o sistema não pode entrar em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados logo no início do desenvolvimento do software.

Importante: É o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas se não forem implementados, o sistema poderá ser implantado e usado mesmo assim.

Desejável: É o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis são requisitos que podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

3.2.2 Requisitos Funcionais

Para o Sistema de Gerenciamento e Distribuição de Disciplinas da FAMAT, foram elicitados os requisitos funcionais detalhados na presente seção através de reuniões com um professor participante da Comissão de Distribuição, e a partir da Resolução 02/2017 FAMAT (Anexo 4) que regulamenta a distribuição de disciplinas naquela unidade.

RF-001 Login: O sistema deve permitir que professores e administradores efetuem o login através do número do seu SIAPE e senha pessoal e conta administradora e senha respectivamente. Prioridade: Essencial.

RF-002 Criar um novo semestre: O sistema deve permitir a criação de um novo período que compreende um semestre letivo, para que seja percorrido todo o ciclo de distribuição de disciplinas. Ao criar o novo semestre, deve apresentar a opção de clonar a estrutura de turmas de semestres anteriores para o novo semestre, sem manter a distribuição das turmas entre o corpo docente. Prioridade: Essencial.

RF-003 Manter disciplinas: O sistema deve gerenciar (inserir, remover, buscar, editar) as disciplinas ofertadas pela FAMAT, para os diversos cursos de graduação da UFU. Deve-se manter um *link* para as ementas, código, nome, carga horária semanal, à qual divisão pertence e à qual unidade acadêmica/curso é destinada. Prioridade: Essencial.

RF-004 Manter docentes: O sistema deve gerenciar (inserir, remover, buscar, editar) os docentes da FAMAT. Devem-se manter: nome, siape, data de admissão, tipo de contrato, divisão à qual ele pertence, email, situação do cadastro no SIGAD, se este é um dos administradores, sua pontuação, a quantidade máxima de horas permitidas para lecionar, o grupo ao qual pertence, se está de licença, se realiza alguma atividade administrativa, se é um assessor administrativo, quais as turmas sob sua responsabilidade e senha de cada professor. Prioridade: Essencial

RF-005 Manter tipos de atividades: O sistema deverá gerenciar (inserir, remover, buscar, editar) os tipos de atividades acadêmicas que podem ser desenvolvidas por docentes. Neste caso, mantêm-se o nome da atividade, o anexo ao qual ela pertence, sua pontuação, se essa pontuação deve ser multiplicada pela quantidade de meses de realização, a validade dos pontos desta atividade, sua descrição. Prioridade: Essencial.

RF-006 Manter atividades realizadas: O sistema deverá gerenciar (inserir, remover, buscar, editar) as atividades acadêmicas realizadas por docentes. É necessário manter o nome da atividade, o anexo ao qual ela pertence, sua pontuação e quem a realizou. Além de manter os seguintes campos de acordo com o anexo:

- Anexo 1: Área e ano do Qualis-CAPES, Agência financiadora.
- Anexo 2: Nenhum campo específico.
- Anexo 3: Horas aulas semanais, N^o de alunos na turma
- Anexo 4: Nenhuma campo específico.

Prioridade: Essencial

RF-007 Classificar docentes: O sistema deverá automaticamente classificar cada docente no momento em que um administrador fechar o período de registro de atividades de acordo com suas atividades realizadas, de modo que os docentes pertençam à um grupo. É realizada uma classificação para cada divisão (Matemática e Estatística). Prioridade: Essencial.

Os grupos serão divididos da seguinte forma:

G3: ficam os professores recém contratados e aqueles que realizaram pelo menos uma atividade de cada um dos anexos 1, 2 e 3, não excedendo 30% do número total do corpo docente, excluindo os docentes com tipo de contrato substituto, temporário e os docentes efetivos de licença. Dentro do grupo a colocação do professor será de acordo com a soma da multiplicação das pontuações das atividades realizadas com a quantidade destas mesmas atividade realizadas.

G2: ficam os professores que realizaram pelo menos uma atividade de pelo menos dois dos anexos 1, 2, e 3, não excedendo 40% do número total do corpo docente,

excluindo os docentes com tipo de contrato substituto, temporário e os docentes efetivos de licença. Dentro do grupo a colocação do professor será de acordo com a soma da multiplicação das pontuações das atividades realizadas com a quantidade destas mesmas atividade realizadas.

G1: ficam os demais professores que não foram classificados em nenhum dos grupos anteriores.

RF-008 Consultar sua posição no ranking do seu grupo: O sistema deve permitir que o professor consulte sua posição no ranking de seu grupo após o período de registro de atividades ter sido encerrado. A identificação dos professores no ranking deve ser feita através do número do seu Siape. Prioridade: Alta

RF-009 Listar disciplinas preferenciais dos docentes: O sistema deverá, em um período especificado, permitir que cada docente efetivo da FAMAT, após tomar consciência de sua classificação, e das disciplinas a serem ofertadas, possa listar e selecionar no mínimo quinze disciplinas que gostaria de ministrar no semestre letivo seguinte e ordená-las por preferência com o auxílio de filtros por turno, curso e dia da semana, e também com um link disponível para conferir a ementa das disciplinas. Prioridade: Essencial

RF-010 Processar atribuição de disciplinas: O sistema deverá, no período especificado, depois de cada docente indicar sua preferência de disciplinas e a partir da indicação de um administrador sobre quais grupos participarão, processar a distribuição efetiva de disciplinas de cada docente, começando pelo docente que obteve maior pontuação no período respectivo. A distribuição se dá em vários turnos, sendo que em cada um apenas uma turma pode ser atribuída a cada docente. Esse ciclo se repete até que todas disciplinas sejam distribuídas ou todos os docentes participantes da distribuição tenham atingido a carga mínima de horas aula semanais. As disciplinas remanescentes serão distribuídas manualmente pelo administrador do sistema. Prioridade: Essencial.

RF-011 Validar atribuições de disciplinas aos docentes: O sistema deverá depois de processar a distribuição de turmas ministradas por cada docente, permitir que um administrador modifique as distribuições, para atender pequenas demandas por parte de docentes, e suas contestações. Prioridade: Importante

RF-012 Esqueceu a senha: O sistema deve permitir que professores solicitem o reset de senha caso a esqueçam utilizando seu siape e email. Prioridade: Essencial

RF-013 Alterar a senha: O sistema deve permitir que professores realizem a alteração de suas senhas pessoais utilizando suas senhas atuais. Prioridade: Baixa

RF-014 Exibir relatório com informações pertinentes: O sistema deve exibir um *dashboard* na tela de início com informações pertinentes ao sistema. Prioridade: Baixa

Administrador: Se o período de registro de atividade está aberto ou fechado; Se o período de seleção de disciplina está aberto ou fechado; Quantos dias faltam para o fechamento do período de registro de atividade; Quantos dias faltam para o fechamento do período de seleção de disciplina; Nº de professores; Nº de disciplinas; % de Professores que realizaram ou não seleção de disciplinas; % de Professores que realizaram ou não o registro de atividades;

Professor: Se o período de registro de atividade está aberto ou fechado; Se o período de seleção de disciplina está aberto ou fechado; Quantos dias faltam para o fechamento do período de registro de atividade; Quantos dias faltam para o fechamento do período de seleção de disciplina; Quantidade de turmas sob sua responsabilidade; Disciplinas mais concorridas; Requisitos Não Funcionais

3.2.3 Requisitos Não Funcionais

Os seguintes requisitos não-funcionais definem como o Sistema de Distribuição de Disciplinas FAMAT deverá funcionar:

RNF-001 Plataforma web: É desejável que o sistema seja desenvolvido em plataforma web, por independência de sistema operacional. Prioridade: Desejável.

RNF-002 Segurança: O sistema deve manter o sigilo das atividades acadêmicas dos professores, bem como sua categoria e disciplinas ministradas. Prioridade: Essencial.

RNF-003 Servidores de aplicação e banco: O sistema deve ser hospedado entre os servidores da Universidade Federal de Uberlândia, com intuito da própria instituição manter o gerenciamento e a disponibilidade do sistema. Prioridade: Desejável.

RNF-04 Rotinas de backup: O servidor de banco deve ter rotinas administrativas, para que o sistema gere backups dos dados com regularidade mensal. Prioridade: Desejável.

RNF-05 Usabilidade: O sistema deve ser intuitivo e fácil de usar independente do nível de familiaridade do usuário com informática. Prioridade: Essencial.

RNF-06 Navegabilidade: Deve ser possível acessar qualquer parte do sistema com no máximo 5 cliques. Prioridade: Desejável.

RNF-07 Aparência: O sistema deve apresentar uma interface amigável e contemporânea. Prioridade: Desejável.

3.3 Arquitetura do Sistema

Baseando-se no padrão de arquitetura MVC definido anteriormente na Seção 2.2, a arquitetura do sistema em questão está representada como mostra a Figura 5. Nota-se então que o *back-end* foi dividido em duas camadas: a *controller* e a *model*. Dentro desta última camada há dois módulos: o módulo de acesso ao banco e o módulo de funções. O primeiro é o que vai criar uma interface de interação entre a aplicação e o banco de dados. E o módulo de funções, por sua vez, contém as regras de negócio e os algoritmos de manipulação de dados. Ambos os módulos são consumidos pela camada *controller*, que recebe as requisições diretamente do usuário através do seu módulo de pontos de acesso *REST*.

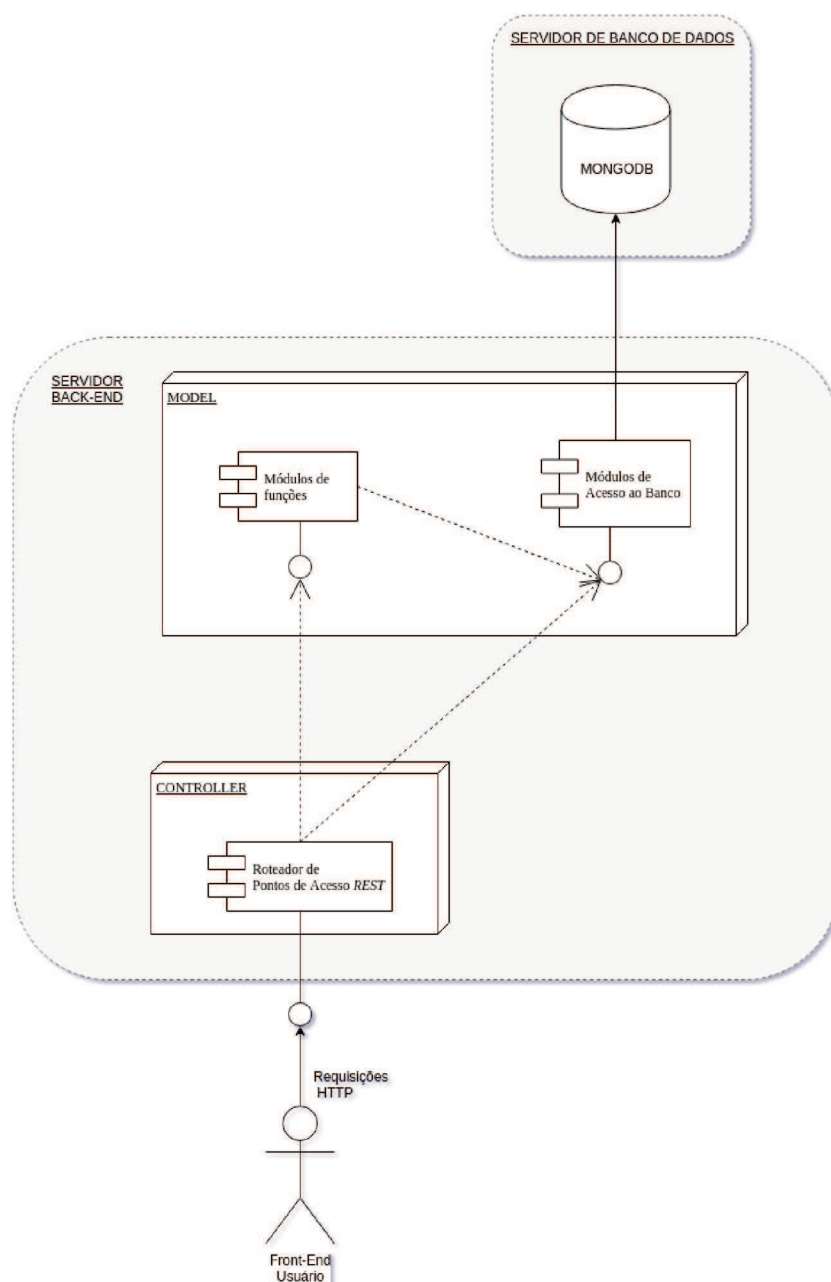


Figura 5 – Arquitetura do sistema.

É importante notar que as tecnologias mencionadas na Seção 3.1 são utilizadas na arquitetura. O *Express JS*, foi utilizado para permitir criar e expôr todos os pontos de acesso para a rede. Já o *NodeJS* é utilizado em todo o *back-end*, como forma de criar um servidor em *JavaScript* que trate todas as requisições e processamentos de forma assíncrona. O *Mongo DB* foi escolhido como sistema gerenciador do banco de dados orientado a documentos. E por fim, o *Angular* é a ferramenta para criar a interface visual, executada no navegador do usuário. Um exemplo das relações existentes entre as camadas e seus módulos é o caso no qual o professor administrador fecha o período de registro de atividades. Como pode-se ver na imagem Figura 6, o usuário realiza uma chamada POST ao *endpoint* disponibilizado na camada *controller*, enviando um objeto de período de seleção com os atributos modificados. Como neste caso de uso o administrador quer fechar o período, ele apenas modifica o atributo *open* para falso. Agora, como o fechamento de período de registro de atividades irá acarretar na atribuição de pontuação e no agrupamento dos professores, o *controller* irá utilizar uma função no módulo de funções da camada *model*: “*closeSelectionPeriod*”. Note que, esta ação irá realizar os cálculos das pontuações acumuladas pelas atividades de cada professor e atribuir um grupo a cada professor.

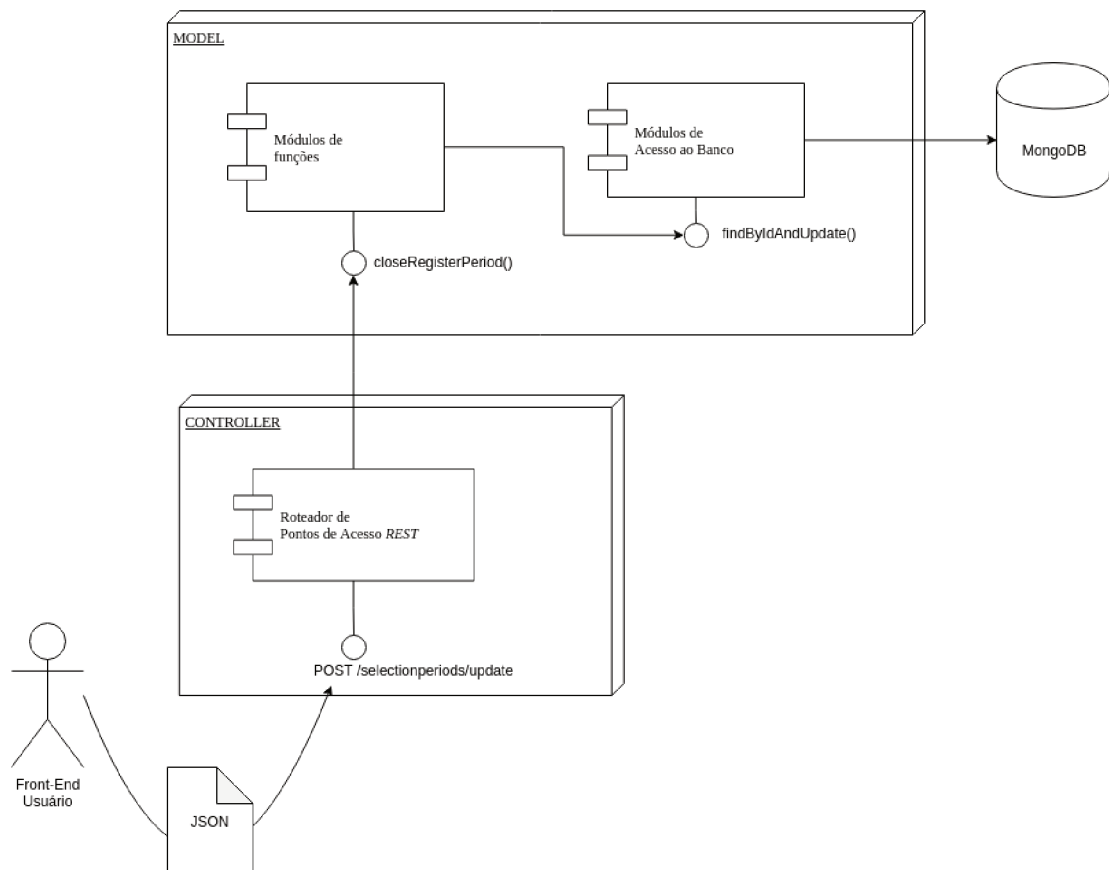


Figura 6 – Interação entre os elementos do sistema para o caso de uso de um fechamento de período de registro de atividades.

3.4 Projeto de Banco de Dados Orientado a Documento

Após uma análise dos requisitos e dos documentos disponibilizados pelo cliente, foi desenvolvido o projeto do banco de dados da aplicação. Observe que sendo o MongoDB o banco utilizado pela aplicação, e o mesmo é orientado a documentos, a arquitetura do banco é reflexo direto dos objetos criados pelo *back-end*. Assim sendo, temos um total de 7 objetos diferentes utilizados pela aplicação. Porém, três deles são parte de um outro objeto, chamado de “Distribuição”. Este objeto engloba os dados dos professores que serão inválidos ao mudar de ano. Por exemplo, as atividades que registraram ou as turmas fornecidas pela faculdade. Portanto, este objeto é identificado apenas pelo ano, e assim podemos manter um histórico de dados para possíveis análises posteriores.

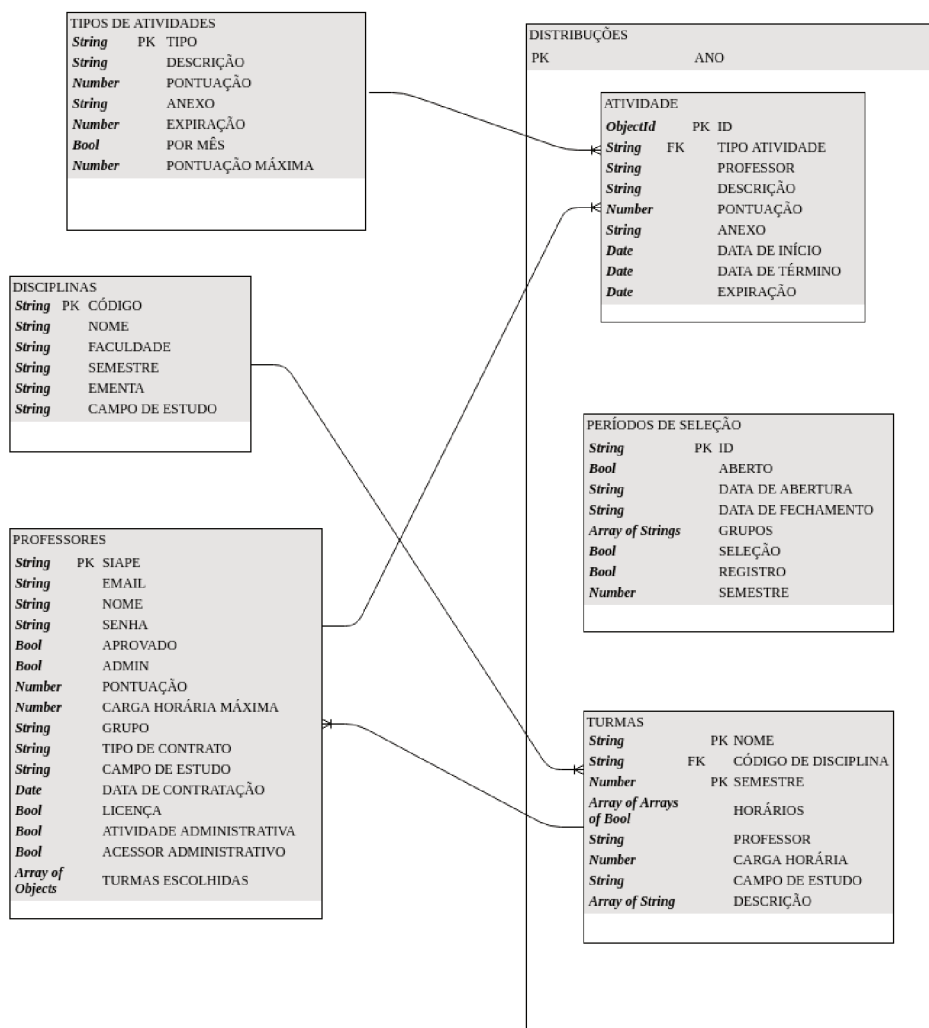


Figura 7 – Objetos no Banco de Dados.

O primeiro objeto atrelado à classificação são as atividades realizadas pelos professores no ano anterior. O segundo objeto incluso é o período de seleção, que representa os períodos específicos, onde certas ações dos professores serão admitidas. E por fim, o último objeto incluso é a turma, uma instância derivada de disciplina e pode a cada semestre ser fornecida pela faculdade em horários diferentes e diversos professores.

O restante dos objetos do projeto do banco de dados são independentes e serão persistidos a cada ano. O primeiro independente é o tipo de atividade, que identifica cada atividade incluída pelo usuário. Esta atividade será usada pelo sistema para atribuir determinado grau de prioridade ao professor durante a distribuição de disciplinas. Outro objeto independente é a disciplina, que é fornecida pela faculdade para, em geral, diversas turmas. E por fim, o objeto usuário, que representa o professor e suas funções e dados identificadores.

3.5 Desenvolvimento SCRUM

3.5.1 Backlog do Produto

A escolha das atividades que compõem o Backlog do Produto deu-se de forma a priorizar os requisitos básicos, como a criação e o registro de disciplinas, docentes e atividades realizadas no semestre. O Backlog do Produto contém:

- RF-001 Login
- RF-002 Criar um novo semestre
- RF-003 Manter Disciplinas
- RF-004 Manter Docentes
- RF-005 Manter Tipos de Atividades
- RF-006 Manter Atividades Realizadas
- RF-007 Classificar Docentes
- RF-008 Consultar sua posição no ranking do seu grupo
- RF-009 Listar Disciplinas Preferenciais dos Docentes
- RF-010 Processar Atribuição de Disciplinas aos Docentes
- RF-011 Validar Processamento de Atribuições de Disciplinas aos Docentes
- RF-012 Esqueceu a senha
- RF-013 Alterar a senha
- RF-014 Exibir relatório com informações pertinentes

3.5.2 Sprint 1

3.5.2.1 Planejamento

Durante o planejamento deste sprint foi decidido que o desenvolvimento iria se orientar pelos requisitos:

- RF-001 Login
- RF-003 Manter Disciplinas
- RF-004 Manter Docentes
- RF-005 Manter Tipos de Atividades
- RF-006 Manter Atividades Realizadas
- RF-013 Alterar a senha

Os requisitos escolhidos para o sprint são em sua maioria referentes ao preenchimento e leitura de dados básicos, que serão a base para a implementação e uso das outras estruturas e funções mais complexas. Dentre as funções mais complexas, destacamos o processamento de pontuação dos professores, que para ser implementada é necessário que haja toda a estrutura de dados, busca, escrita de informações dos docentes e das atividades, assim como dos períodos de registro de atividades que são controlados pelo administrador.

3.5.2.2 Execução

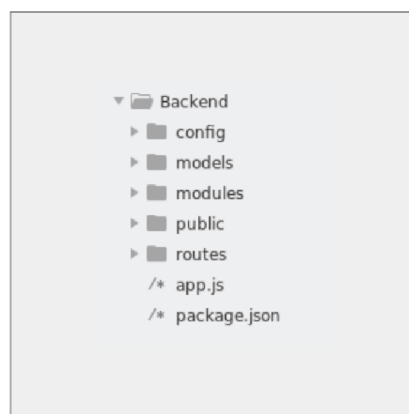


Figura 8 – Organização dos diretórios de código da aplicação.

A organização determinada dos códigos e módulos foi definida como mostra a Figura 8. No diretório “models” estão as definições dos objetos das entidades de dados, e as funções de acesso ao banco. Em “config” estão as configurações de acesso de rede ao MongoDB, como endereço, porta, usuário de acesso, senha e nome do *database*. E na pasta

“routes” estão as declarações de todas as APIs REST. E por fim, em “modules”, serão salvos os módulos de cada entidade onde são declaradas as funções de manipulação de dados e implementação de regras de negócio que serão consumidas internamente conforme a necessidade das APIs ou até entre os módulos.

O arquivo 3.1 é o arquivo central de inicialização e configuração do servidor. É onde estão definidas todas as rotas até as APIs, a porta do servidor por onde as requisições do *front-end* irão chegar, e é onde se faz a instanciação da conexão com o banco de dados. A linha 2 determina a requisição de importação do framework *Express*, que foi descrito na seção sobre a pilha MEAN. Com o framework importado, na linha 23 instancia-se um novo objeto express com o nome “app”. Em seguida, definem-se várias configurações através da função “use”, como nas linhas entre 62 e 68. Estas configurações fazem o redirecionamento das requisições web para as APIs declaradas nos arquivos da pasta “routes”. Com todas as devidas definições de rotas no *Express*, é na linha 86 onde se inicializa o serviço de servidor atrelado a uma porta de rede da máquina.

Listagem 3.1 – Arquivo central de configuração e icialização do servidor back-end.

```

1 const swaggerJSDoc = require('swagger-jsdoc')
2 const express      = require('express');
3 const path         = require('path');
4 const bodyParser   = require('body-parser');
5 const cors         = require('cors');
6 const passport     = require('passport');
7 const mongoose     = require('mongoose');
8 const config       = require('./config/database');
9 const packageJson = require('./package');
10
11
12 const DIRNAME = typeof __dirname !== 'undefined' ? __dirname :
    FILENAME.replace(/[\]\\\/\.\*\?$/ , '');
13
14 mongoose.connect(config.database,{ useNewUrlParser: true });
15 mongoose.connection.on('Connected',()=>{
16   console.log('Connected to database'+config.database);
17 });
18 mongoose.connection.on('error',(err)=>{
19   console.log('Database error : '+ err);
20 });
21
22
23 const app      = express();
24 const port     = 3000;
25
26 const users = require('./routes/users');
27 const subjects = require('./routes/subjects');
```

```
28 const classes = require('./routes/classes');
29 const selectionperiods = require('./routes/selectionperiods');
30 const activitytypes = require('./routes/activitytypes');
31 const activities = require('./routes/activities');
32 const distributions = require('./routes/distributions');
33
34
35 const swaggerDefinition = {
36   info: {
37     title: packageJson.name,
38     version: process.env.APP_VERSION,
39     description: packageJson.description
40   },
41   host: '0.0.0.0:3000',
42   basePath: '/'
43 }
44
45
46 var swaggerOptions = {
47   swaggerDefinition: swaggerDefinition,
48   apis: ['./routes/*.js']
49 };
50
51 const swaggerSpec = swaggerJSDoc(swaggerOptions)
52
53
54 app.use(cors());
55 app.use(express.static(path.join(DIRNAME, 'public')));
56 app.use(bodyParser.json());
57
58 app.use(passport.initialize());
59 app.use(passport.session());
60 require('./config/passport')(passport);
61
62 app.use('/users', users);
63 app.use('/subjects', subjects);
64 app.use('/classes', classes);
65 app.use('/selectionperiods', selectionperiods);
66 app.use('/activitytypes', activitytypes);
67 app.use('/activities', activities);
68 app.use('/distributions', distributions);
69
70
71 app.get('/', (req, res) => {
72   res.send('Invalid endpoint');
73 });
74
```

```
75 app.get('/swagger.json', (req, res) => {
76   res.setHeader('Content-Type', 'application/json');
77   res.send(swaggerSpec);
78 })
79
80 app.use('/docs/', express.static('./public/api-docs/'));
81
82 app.get('*', (req, res) => {
83   res.sendFile(path.join(DIRNAME, 'public/index.html'));
84 });
85
86 app.listen(port, () => {
87   console.log('Server started on port '+port);
88 });
```

Na execução destes requisitos, foi criada inicialmente a arquitetura básica do usuário, que é a entidade que representa os docentes e os administradores. Tratando do requisito RF-004, o código 3.2 define o objeto usuário e a estrutura que vai ser salva no banco de dados. Determinam-se então os atributos que descrevem o usuário, e seus tipos de dado.

Listagem 3.2 – Modelo do Usuário

```
1 const mongoose = require('mongoose');
2 const bcrypt   = require('bcryptjs');
3 const config   = require('../config/database');
4
5 // USER SCHEMA
6 const userSchema = mongoose.Schema({
7   name: {
8     type: String
9   },
10  email: {
11    type: String,
12    required: true
13  },
14  siape: {
15    type: String,
16    required: true
17  },
18  password: {
19    type: String,
20    required: true
21  },
22  approved: {
23    type: Boolean,
24    required: true
```

```
25  },
26  admin: {
27    type: Boolean,
28    required: true
29  },
30  score: {
31    type: Number,
32    required: true
33  },
34  maxHours: {
35    type: Number,
36    required: true
37  },
38  group: {
39    type: String,
40    required: false
41  },
42  contractType: {
43    type: String,
44    required: true
45  },
46  field: {
47    type: String,
48    required: true
49  },
50  hiringDate: {
51    type: Date,
52    required: true
53  },
54  workLeave: {
55    type: Boolean,
56    required: true
57  },
58  administrativeActivity: { // Coordenadores e afins
59    type: Boolean,
60    required: true
61  },
62  administrativeAccessor: { // Acessor administrativo
63    type: Boolean,
64    required: true
65  },
66  chosenClasses:[{
67    name: {
68      type: String,
69      required:false,
70    },
71    subjectCode: {
```

```

72     type: String,
73     required: false,
74   },
75   semester: {
76     type: Number,
77     required: false
78   }
79   }]
80 });
81
82 const User = module.exports = mongoose.model('User', userSchema);

```

Em conjunto com a declaração da entidade, é necessário que se declare também as operações básicas mais primitivas e essenciais a serem realizadas diretamente com o banco de dados. Como por exemplo, as funções *getUsersByGroup* que traz uma lista de todos os usuários no banco de dados que fazem parte de um certo grupo especificado por parâmetro, e *updateUser* que sobrescreve um objeto no banco com outro usuário passado por parâmetro, usando como referência o *hash* do identificador (id). Todas as funções citadas anteriormente, bem como outras funções essenciais também foram escritas no arquivo 3.3.

Listagem 3.3 – Modelo do Usuário

```

1 module.exports.getUserById = function(id, callback){
2   User.findById(id, callback);
3 }
4
5 module.exports.getUserBySiape = function(siape, callback){
6   const query = {siape: siape};
7   User.findOne(query, callback);
8 }
9
10 module.exports.addUser = function(newUser, callback){
11   bcrypt.genSalt(10, (err, salt) => {
12     bcrypt.hash(newUser.password, salt, (err, hash) => {
13       if(err) throw err;
14       newUser.password = hash;
15       newUser.save(callback);
16     });
17   });
18 }
19
20 module.exports.deleteUser = function(targetUser, callback){
21   User.findByIdAndRemove(targetUser.id, callback);
22 }
23
24 module.exports.updateUser = function(modifiedUser, callback){

```

```
25 User.findByIdAndUpdate(modifiedUser._id,modifiedUser,callback);
26 }
27
28 module.exports.comparePassword = function(candidatePassword, hash,
      callback){
29   bcrypt.compare(candidatePassword, hash, (err, res) => {
30     if(err) throw err;
31     callback(null, res);
32   });
33 }
34
35 module.exports.getBysiape = function(siapeRcvd,callback){
36   const query = {siape: siapeRcvd};
37   const options = {__v:0};
38   User.find(query,options,callback);
39 }
40
41 module.exports.getAllUsers = function(callback){
42   const query = {};
43   const options = {};
44   User.find(query,options,callback);
45 }
46
47 module.exports.getUserScoreList = function(callback){
48   User.aggregate([{$match:{workLeave:false,approved:true}},{$sort:{
      score:-1}},{$project:{}}],callback);
49 }
50
51 module.exports.getUserScoreListGroup = function(rcvdgroup,callback)
      {
52   User.aggregate([{$match:{group:rcvdgroup,workLeave:false,approved:
      true}},{$sort:{score:-1}}],callback);
53 }
54
55 module.exports.getUsersByGroup = function(groupRcvd,callback){
56   const query = { group: groupRcvd ,workLeave: false ,approved: true
      };
57   User.find(query,{},callback);
58 }
59
60 module.exports.getUsersByGroupAndField = function(groupRcvd,fieldRcvd
      ,callback){
61   const query = {group: groupRcvd , field: fieldRcvd, workLeave:
      false ,approved: true, contractType:"Fixo"};
62   User.find(query,{},callback);
63 }
64
```

```
65 module.exports.getByField = function(fieldRcvd,callback){
66   const query = { field: fieldRcvd, workLeave: false ,approved: true ,
        contractType:"Fixo"};
67   User.find(query,{},callback);
68 }
```

Algumas destas funções são necessárias pelo *front-end*, por isso foi necessário externalizá-las diretamente nas APIs sem uso de outras funções. Assim, podemos ver um trecho do arquivo em 3.4, em que a declaração de alguns pontos de acesso REST que receberão as requisições da rede.

Vemos a declaração de uma API para o cadastro de novos usuários na linha 3, que utiliza a requisição do tipo POST para receber dados em formato JSON, e determina que o caminho desta função na URL será composta por “/register”. Porém, para sabermos o caminho todo desta API devemos observar a configuração realizada no arquivo 3.1 na linha 26 e linha 62, onde o servidor carrega o arquivo para uma variável e faz um roteamento das APIs ali contidas para “/users”. Assim podemos ver que o caminho todo resultante se torna: “<ip-servidor>:<porta>/users/register”.

Listagem 3.4 – Algumas declarações de APIs RESTful da entidade Usuário

```
1 const SelectionPeriod = require('../models/selectionperiod');
2
3 router.post('/register',(req,res,next)=>{
4   if(req.body.admin){
5     admin = req.body.admin;
6   }
7   else{
8     admin = false;
9   }
10
11   if(req.body.hiringDate){
12     hiringdate = new Date(req.body.hiringDate);
13     console.log(req.body.hiringDate + "\n" + hiringdate);
14   }
15
16   let newUser = new User({
17     name : req.body.name,
18     email : req.body.email,
19     siape : req.body.siape,
20     password: req.body.password,
21     group : req.body.group,
22     contractType: req.body.contractType,
23     field : req.body.field,
24     approved: false,
25     admin : admin,
```

```
26     score : 0,
27     maxHours: 0,
28     chosenClasses: [],
29     hiringDate: hiringdate,
30     workLeave : req.body.workLeave,
31     administrativeActivity: req.body.administrativeActivity,
32     administrativeAccessor: req.body.administrativeAccessor
33   });
34
35   User.getUserBysiape(req.body.siape, (err, user_searched) => { // **
36     if(err) res.json({success: false, msg: 'Error: '+err});
37     else if(user_searched){
38       return res.json({success: false, msg: 'siape already registered
39         ! ' + err});
40     }
41     else{
42       User.addUser(newUser,(err,user) => {
43         if(err){
44           res.json({success: false, msg:'Failed to register user:' +
45             err});
46         } else{
47           res.json({success: true, msg:'User registered!'});
48         }
49       });
50     }
51   });
52
53   router.post('/delete',(req,res,next)=>{
54
55     let targetUser = {
56       id : req.body.id
57     };
58
59     User.deleteUser(targetUser,(err, result)=>{
60       if(err){
61         res.json({success: false, msg: 'Failed to delete user :'+err});
62       } else{
63         res.json({success: true, msg: 'User deleted!'});
64       }
65     });
66   });
67
68   router.post('/update',(req,res,next)=>{
69     let modifiedUser = req.body;
70
```

```

71   User.updateUser(modifiedUser,(err, result)=>{
72     if(err){
73       res.json({success: false, msg: 'Failed to update user : '+err})
74       ;
75     } else{
76       res.json({success: true, msg: 'User updated!'});
77     }
78   });
79
80
81
82 module.exports = router;

```

Por outro lado a funcionalidade de alteração de senhas de um usuário é uma operação mais complexa exigida pelo *front-end* e necessita de uma manipulação maior de dados. Para tal funcionalidade (que é exigida pelo requisito RF-013, deste sprint) declaramos uma API para receber a requisição HTTP pela rede, com os dados necessários (demonstrado na listagem 3.5) e uma função no módulo do usuário, conforme em 3.6.

Listagem 3.5 – Alteração de senha do usuário: API

```

1 router.post('/changepassword',(req,res,next)=>{
2   UserModule.changeUserPassword(req.body.siape,req.body.email,req.
      body.newpwd,(err,result)=>{
3     if(err) res.json({success: false, msg: 'Erro: '+err.message});
4     else{
5       res.json(result);
6     }
7   });
8 })

```

Listagem 3.6 – Alteração de senha do usuário: Função do Módulo de Entidade Usuário

```

1
2   changeUserPassword: function(siape,email,newpwd,callback){
3     User.getUserBysiape(siape, (err, user_searched) => {
4       if(err) callback(Error(""+err));
5       else {
6         if(user_searched){
7           if(email == user_searched.email){
8             bcrypt.genSalt(10, (err,salt) => {
9               bcrypt.hash(newpwd, salt, (err, hash) => {
10                 if(err) callback(Error(""+err));
11                 else{
12                   User.updateUser({"_id": ObjectId(user_searched._id)
                      , password: hash}),(err,result)=>{

```

```
13             if(err) callback(Error(""+err));
14             else callback(null,true);
15         });
16     }
17     });
18     });
19     } else callback(Error ('Email and siape not compatible'));
20 }
21 else{
22     callback(Error ('Cannot find the user using given siape'));
23 }
24 }
25 });
26 },
```

Pode-se notar então que para realizar a alteração de senha, a função irá receber por parâmetro o SIAPE do docente (que o identifica, conforme mostrado na arquitetura 7), o seu email e a nova senha. Assim sendo, primeiramente se faz a busca das informações completas do usuário ao MongoDB e, caso os dados recebidos sejam verdadeiros e coerentes, o usuário vai ser encontrado, importado do Mongo e armazenado na variável “user_searched”. Com os dados do usuário faz-se então uma conversão da senha para *hash*, antes de enviar para o banco de dados.

Quanto ao cadastro de novos usuários, este foi declarado a API “/register” exibida na listagem 3.7. Tal cadastro exige que o *front-end* envie determinadas informações sobre o usuário a ser registrado no banco, bem como o nome, SIAPE, email, senha, tipo de contratação, campo de estudo, data de contratação, se está de licença, se cumpre alguma função administrativa e se é acessor administrativo. A partir destas informações um objeto “newUser” é criado, e nele serão preenchidas mais algumas informações que são comuns a todo usuário recém-criado.

Todos os novos usuários são tidos como não aprovados até que o seu cadastro tenha sido aprovado por um administrador do sistema. Por isso, o objeto do novo usuário recebe um atributo “approved” e seu valor é “false”, como vemos na linha 22. É importante notar também que se faz uma consulta ao banco para verificar se o usuário que solicita cadastramento já está no sistema antes de cadastrá-lo de fato, a partir da linha 33.

Listagem 3.7 – API para o cadastro de um novo usuário.

```
1 router.post('/register',(req,res,next)=>{
2   if(req.body.admin){
3     admin = req.body.admin;
4   }
5   else{
6     admin = false;
```

```

7   }
8
9   if(req.body.hiringDate){
10    hiringdate = new Date(req.body.hiringDate);
11    console.log(req.body.hiringDate + "\n" + hiringdate);
12  }
13
14  let newUser = new User({
15    name    : req.body.name,
16    email   : req.body.email,
17    siape   : req.body.siape,
18    password: req.body.password,
19    group    : req.body.group,
20    contractType: req.body.contractType,
21    field    : req.body.field,
22    approved: false,
23    admin    : admin,
24    score    : 0,
25    maxHours: 0,
26    chosenClasses: [],
27    hiringDate: hiringdate,
28    workLeave : req.body.workLeave,
29    administrativeActivity: req.body.administrativeActivity,
30    administrativeAccessor: req.body.administrativeAccessor
31  });
32
33  User.getUserBysiape(req.body.siape, (err, user_searched) => { // **
34    if(err) res.json({success: false, msg: 'Error: '+err});
35    else if(user_searched){
36      return res.json({success: false, msg: 'siape already registered
37        ! ' + err});
38    }
39    else{
40      User.addUser(newUser, (err, user) => {
41        if(err){
42          res.json({success: false, msg: 'Failed to register user: '+
43            err});
44        } else{
45          res.json({success: true, msg: 'User registered!'});
46        }
47      });
48    }
49  });

```

Para a criação das funcionalidades de edição, remoção e inserção do usuário, de disciplina, tipo de atividade e atividade, foram desenvolvidos seguindo o mesmo padrão

do que foi demonstrado. Portanto, se cumprem as funcionalidades exigidas pelo requisitos mencionados no backlog do sprint.

3.5.2.3 Revisão do Sprint

Todas as operações básicas de inserção e busca dos objetos usuário, tipo de atividade, atividade e disciplinas foram desenvolvidos e funcionam. Assim como a funcionalidade de alteração de senha e de autenticação.

3.5.3 Sprint 2

3.5.3.1 Backlog

Durante o planejamento deste sprint foi decidido que o desenvolvimento iria se orientar pelos requisitos:

- RF-007 Classificar Docentes
- RF-008 Consultar sua posição no ranking do seu grupo

3.5.3.2 Execução

Pela quantidade de passos envolvidos a fim de distribuir as turmas pelos professores, todo o fluxo geral foi dividido em duas fases (como ilustrado na Figura 9). A primeira fase que contém as etapas referentes à pontuação e divisão em grupos foi escolhida para ser desenvolvida neste segundo sprint. A implementação então para este sprint concentra-se nos requisitos específicos referentes à primeira fase.

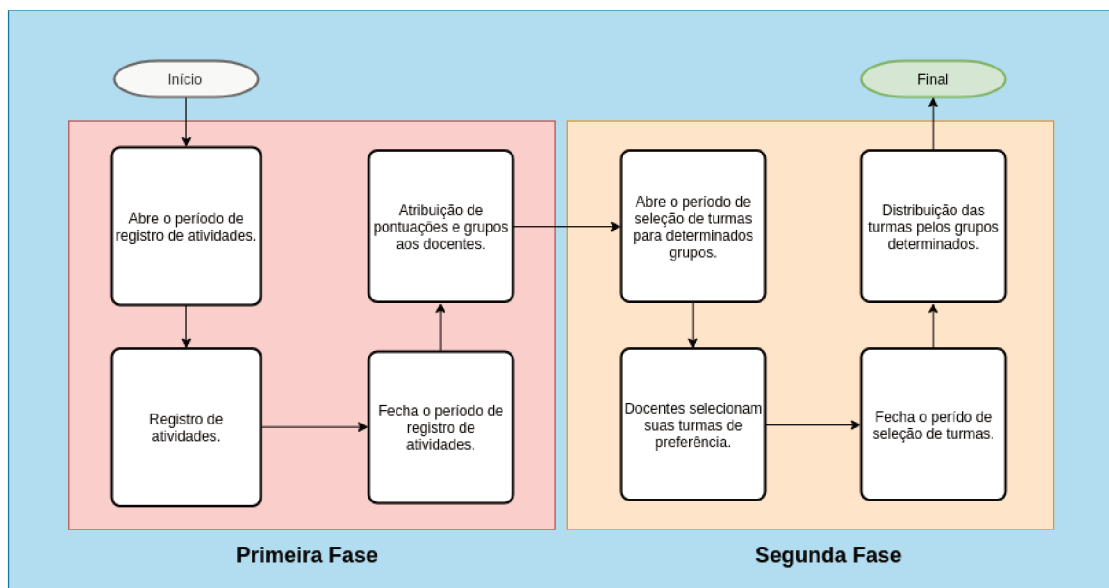


Figura 9 – Visão Geral do Processo de Distribuição de Disciplinas.

Assim, os passos citados na Figura 9 contêm outros procedimentos menores envolvidos. Para classificar os docentes por exemplo, é necessário desenvolver as funções para computar o registro de atividades de um docente. Ou seja, validar e atribuir sua devida pontuação e o inserir no grupo devido. O que deve ser processado observando a data de expiração de cada atividade, limite máximo de pontos acumuláveis por docente, e a sua duração se necessário. Também é preciso desenvolver a forma pela qual o administrador irá controlar os períodos de registro de atividade (até mesmo de seleção de disciplinas).

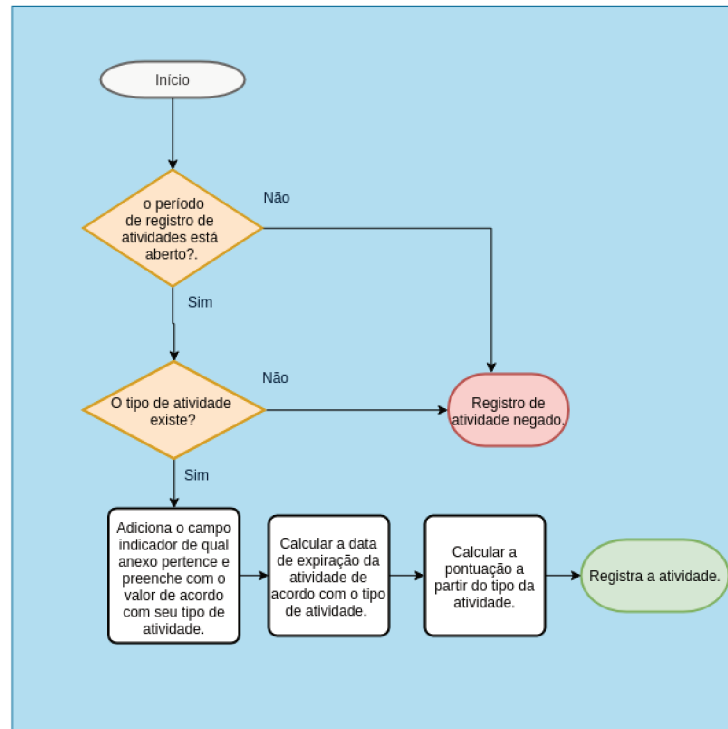


Figura 10 – Fluxo do Registro de Atividades dos Docentes.

Na Figura 10 é possível visualizar as verificações envolvidas que são necessárias quando o sistema recebe uma requisição do usuário para o registro de alguma atividade. Com a primeira verificação, assegura-se que o docente pode fazer registro de atividades apenas nos períodos onde o administrador do sistema determinar. No código a implementação deste período de seleção se deu pela criação de uma entidade chamada “*Selection Period*”, que está contida dentro da entidade maior “*Distribution*” conforme mostrado na Figura 7. Seus atributos foram determinados de tal forma que é possível utilizar a mesma entidade para ambos os períodos de seleção de turmas e o de registro de atividades. Tal identidade se dá pelos atributos “register” e “selection” (linhas 22 e 21 da listagem 3.8).

Listagem 3.8 – Modelo do Período de Seleção

```

1  selectionPeriod: [{
2    open: {
3      type: Boolean,
4      required: true

```

```
5     },
6     closureDate: {
7       type: String,
8       required: false
9     },
10    openingDate: {
11      type: String,
12      required: false
13    },
14    groups: [{
15      type: String,
16      required: true
17    }],
18    selection: {
19      type: Boolean,
20      required: true
21    },
22    register: {
23      type: Boolean,
24      required: true
25    },
26    semester: {
27      type: Number,
28      required: true
29    }
30  }]
```

Seguindo para a segunda verificação no fluxograma 10 tem-se a validação do tipo de atividade da atividade a ser registrada. Se este não existir, significaria que a atividade não se encaixa em nenhum padrão identificável, e por tanto não seria possível fazer qualquer tipo de pontuação. Por isso a atividade com o tipo inexistente não seria aceita.

Listagem 3.9 – Modelo do Tipo de Atividades

```
1 const activityTypeSchema = mongoose.Schema({
2   type: {
3     type: String,
4     required: true
5   },
6   description: {
7     type: String,
8     required: true
9   },
10  score: {
11    type: Number,
12    required: true
13  },
```

```

14  attachment: {
15      type: String,
16      required: true
17  },
18  expiration: {
19      type: Number,
20      required: true
21  },
22  byMonth: {
23      type: Boolean,
24      required: true
25  },
26  maxScore: {
27      type: Number,
28      required: true
29  }
30 });

```

Por outro lado, com um tipo válido a atividade apenas precisa de receber algumas informações para ser registrada. A primeira delas é identificação de qual anexo pertence, para facilitar a atribuição de grupos de cada docente posteriormente. Observe que sem esta informação na própria atividade, seria necessário buscar o tipo de atividade no banco de dados de cada atividade a ser verificada. Outra informação a ser recebida é uma data de expiração, que é calculada como a soma da data de término da atividade com o atributo “expiration” do tipo de atividade (listagem 3.9 linha 18).

Este atributo representa a quantidade de semestres nos quais as atividades deste tipo serão válidas. Após estes dados serem inseridos na atividade a ser registrada, a pontuação precisa ser calculada. E esta será determinada por unidade de atividade, ou pela quantidade de meses que durou. Para identificar o tipo de cálculo usa-se o atributo “byMonth” na entidade do tipo de atividade, conforme pode ser visto na listagem 3.9, linha 22. Quando o administrador do sistema fechar o período de registro de atividades, os docentes perderão a capacidade de registrar mais atividades e o sistema irá calcular as suas pontuações e os dividirá pelos grupos. Todas as verificações e processos realizados quando há a requisição para o fechamento foram divididos entre estas duas partes, que estão demonstradas nas Figuras 11 e 12.

Listagem 3.10 – Função Para o Fechamento do Período de Registro de Atividades.

```

1  closeRegisterPeriod: function(year,modifiedSelectionPeriod, callback)
2      {
3      UserModule.setUsersTotalScore(year,(err,result)=>{
4          if(err) callback(Error("(set_users_totalsocore)"+err),false);
5          else{

```

```

6      UserModule.setAllUsersGroup((err,result)=>{
7          if(err) callback(Error("(set_all_users_group)" + err), false);
8          else {
9              callback(err, true);
10         }
11     });
12 }
13 });
14 },

```

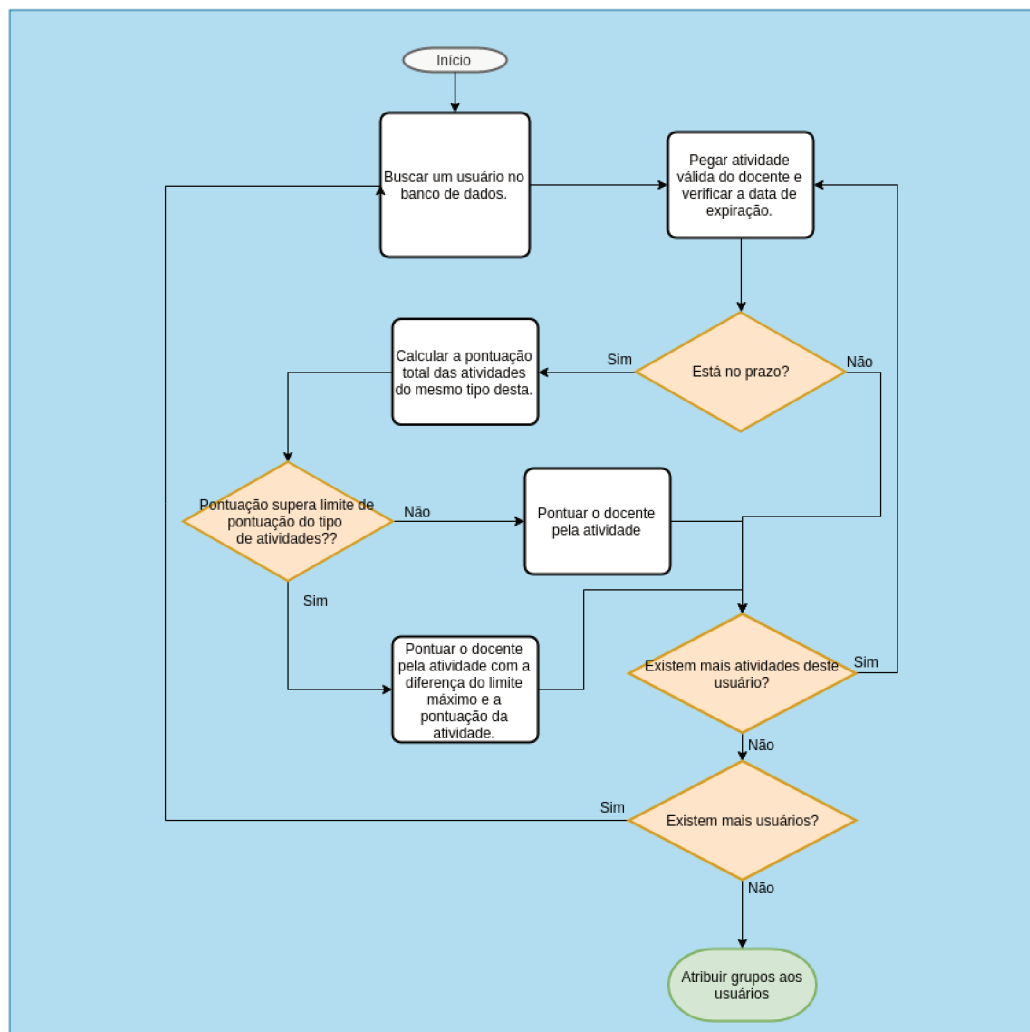


Figura 11 – Fluxo do Fechamento do Período de Registro de Atividades: Atribuição de pontos

Para atribuir as pontuações devidas para os docentes, o sistema deverá buscar em meio a todas as atividades registradas aquelas que ainda estão dentro do prazo de validade. Uma vez que estas foram encontradas a pontuação será usada na pontuação do usuário. Caso o administrador tenha estipulado uma pontuação máxima permitida no tipo de atividade da atividade em questão, o docente não irá receber quaisquer pontos além deste valor.

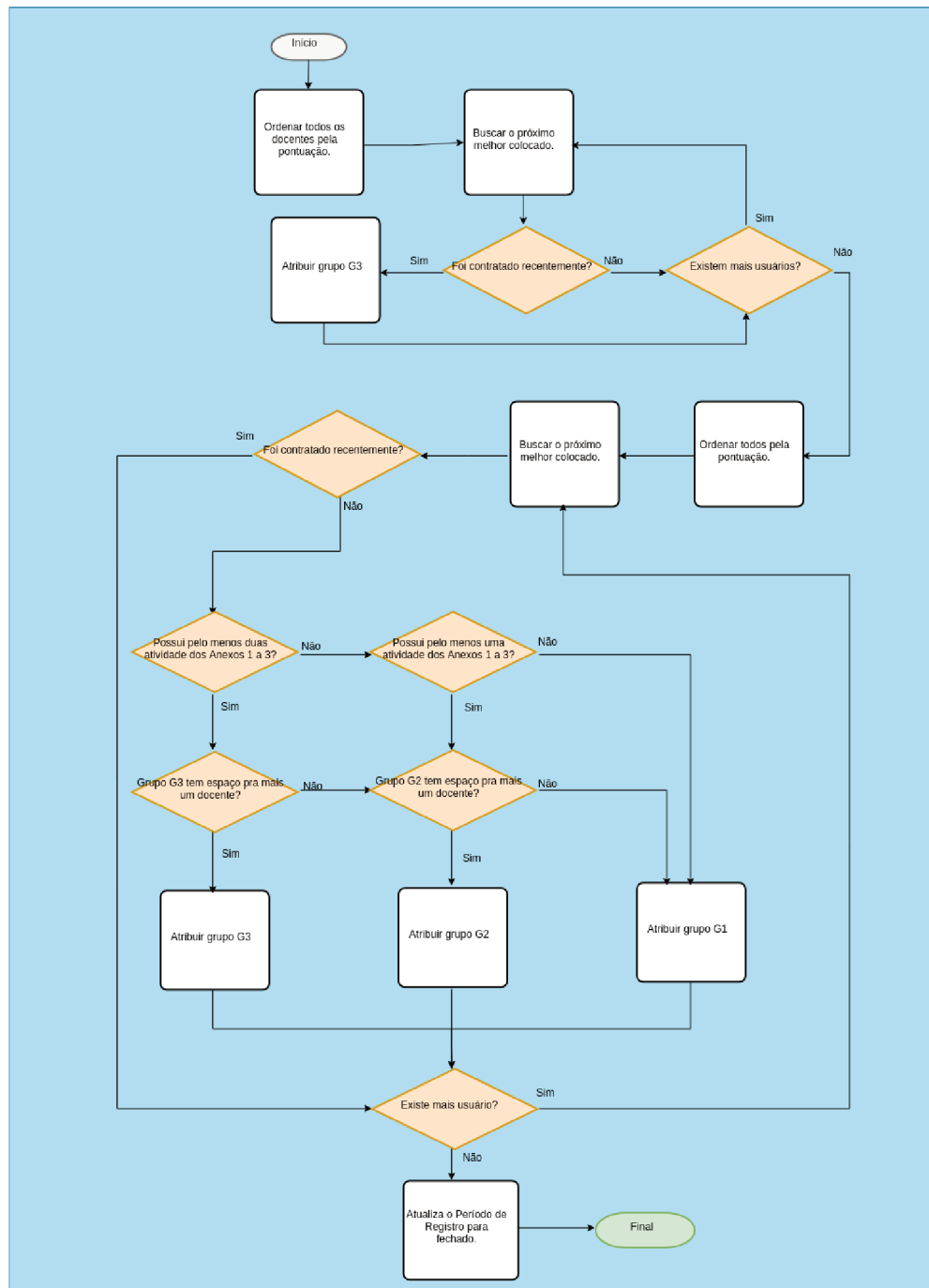


Figura 12 – Fluxo do Fechamento do Período de Registro de Atividades: Atribuição de grupos

Para atribuir um grupo ao docente, segundo as regras do arquivo da FAMAT em anexo (Artigo 4º), deve-se considerar a pontuação de cada usuário. Portanto, o primeiro procedimento a ser realizado é o de ordenar todos os docentes (que não estão de licença e são efetivos) pela pontuação. Existe ainda outro procedimento que precede a atribuição de grupos de fato, que é a atribuição dos usuários recém contratados ao grupo G3 nas últimas posições. Por isso eles serão atribuídos aos grupos anteriormente, fazendo assim com que o cálculo de quantos docentes ainda podem entrar neste grupo sejam feitos ao

decorrer de todo o processo.

Com os recém contratados atribuídos ao G3 percorre-se a lista de todos os docentes ordenados pela pontuação, de forma decrescente. Para cada um é feita a análise sobre os anexos das atividades registradas pelo professor, segundo o que é determinado também no artigo 4º da resolução em anexo e foi descrito na seção 3.1.

3.5.3.3 Revisão do Sprint

Durante este sprint foi possível concluir a primeira fase do processo de distribuição, que está representada pelo requisito funcional RF-007, referente a todo o processo de registro de atividades e classificação dos docentes. O restante ficou pendente, para ser desenvolvido no próximo sprint.

3.5.4 Sprint 3

3.5.4.1 Backlog

Durante o planejamento deste sprint foi decidido que o desenvolvimento iria se orientar pelos requisitos:

- RF-002 Criar um novo semestre
- RF-009 Listar Disciplinas Preferenciais dos Docentes
- RF-010 Processar Atribuição de Disciplinas aos Docentes
- RF-011 Validar Processamento de Atribuições de Disciplinas aos Docentes

3.5.4.2 Execução

Para o desenvolvimento do requisito RF-002 é possível considerar vários casos de uso por parte do administrador. Este pode decidir criar um semestre inteiramente novo sem turmas registradas, ou então criar uma cópia de algum semestre anterior com as turmas sem professores atribuídos. Para o primeiro, caso o novo semestre seja de um ano diferente, o *front-end* poderá utilizar do ponto de acesso mostrado na listagem 3.11. Quando o novo semestre for vazio e no mesmo ano atual em uso no sistema, não há a necessidade de qualquer processamento. Isto se deve ao fato de que as atividades, que são parte do objeto maior “Distribuição”, possuem um identificador de semestre em si. Para o caso no qual o administrador quiser copiar as turmas de um semestre a outro diferente foi desenvolvida uma função de clonagem. Esta recebe como entrada o ano e semestre de origem, e o ano e semestre de destino.

```
1 router.post('/register',(req,res,next)=>{
2   var selectionPeriod = [];
3   if(req.body.selectionPeriod.length == 0){
4     selectionPeriod = [
5       {
6         open          : false,
7         closureDate   : "",
8         openingDate   : "",
9         groups        : [],
10        selection      : false,
11        register       : true,
12        semester       : -1
13      },
14      {
15        open          : false,
16        closureDate   : "",
17        openingDate   : "",
18        groups        : [],
19        selection      : true,
20        register       : false,
21        semester       : 1
22      },
23      {
24        open          : false,
25        closureDate   : "",
26        openingDate   : "",
27        groups        : [],
28        selection      : true,
29        register       : false,
30        semester       : 2
31      }
32    ]
33  }
34
35  let newDistribution = new Distribution({
36    year      : req.body.year,
37    class     : req.body.class,
38    selectionPeriod : selectionPeriod,
39    activity   : req.body.activity
40  });
41
42  Distribution.addDistribution(newDistribution,(err,distribution) =>
43    {
44      if(err){
45        res.json({success: false, msg:'Failed to register distribution
46          '+err});
47      } else{
```

```
46     res.json({success: true, msg: 'Distribution registered!'});  
47   }  
48   });  
49 });
```

Quanto ao requisito RF-009 foi necessário criar um ponto de acesso onde o usuário poderia escolher uma turma. A implementação da escolha de turmas envolveu algumas verificações e procedimentos que estão ilustrados no fluxograma 13. Das verificações é importante notar que o professor não poderá selecionar quaisquer turmas se seu grupo não estiver incluso na lista de grupos permitidos para o atual período de seleção. Também não poderá selecionar caso o mesmo esteja de licença. Segundo o artigo 17 da resolução (Anexo 4), a escolha da turma só pode ser efetivada caso a disciplina tenha uma carga horária semanal maior ou igual ao mínimo permitido pelo grupo ao qual o docente fora introduzido, por isso tal verificação também é realizada.

Em seguida, realizou-se a codificação do requisito que trata da atribuição das turmas em si, o que foi implementado está ilustrado pela Figura 14.

Quando o administrador abre o período de seleção de turmas, ele escolhe quais grupos irão fazer a seleção e quais irão fazer parte dos turnos de atribuição. O sistema então irá realizar a distribuição passando por cada docente, seguindo a ordem de prioridade dos grupos “G3 > G2 > G1”, e atribuindo uma turma por usuário. Assim que todos os usuários forem processados, retorna-se para o topo da lista e o procedimento é realizado por mais algumas vezes. Cada repetição destas é chamada de “turno”.

A quantidade de turnos precisava ser determinada antes da realização da distribuição. Para isso fora utilizado um cenário de pior caso, onde não houvesse limite de carga horária por docente e cada docente escolhesse todas as turmas diferentes uns dos outros. Fazendo assim com que cada usuário processado acarrete em uma atribuição efetiva de turma. Tal cenário pode ser visualizado na Figura 15.

Assim sendo, o menor número de turnos necessários para atribuição de todas as turmas escolhidas para o pior caso é igual à quantidade de turmas da maior lista de escolhas dentre os docentes. Uma vez que o sistema calculou a quantidade de repetições, a distribuição acontece. A análise a ser feita em uma turma na lista das escolhas de um docente é composta por várias verificações conforme Figura 14. A primeira delas é a checagem de disponibilidade da turma, ou seja, se ela já não foi atribuída a algum outro docente. A segunda verificação é referente ao artigo 16 da referida Resolução (Anexo 4), que determina uma carga horária mínima para cada docente de determinado grupo.

Feita a devida análise sobre a turma de determinado docente, e aprovada, a carga horária desta será subtraída do atributo controlador da quantidade de horas semanais do máximo permitido pelo docente (“maxHours”). Assim, a turma pode então ser atribuída

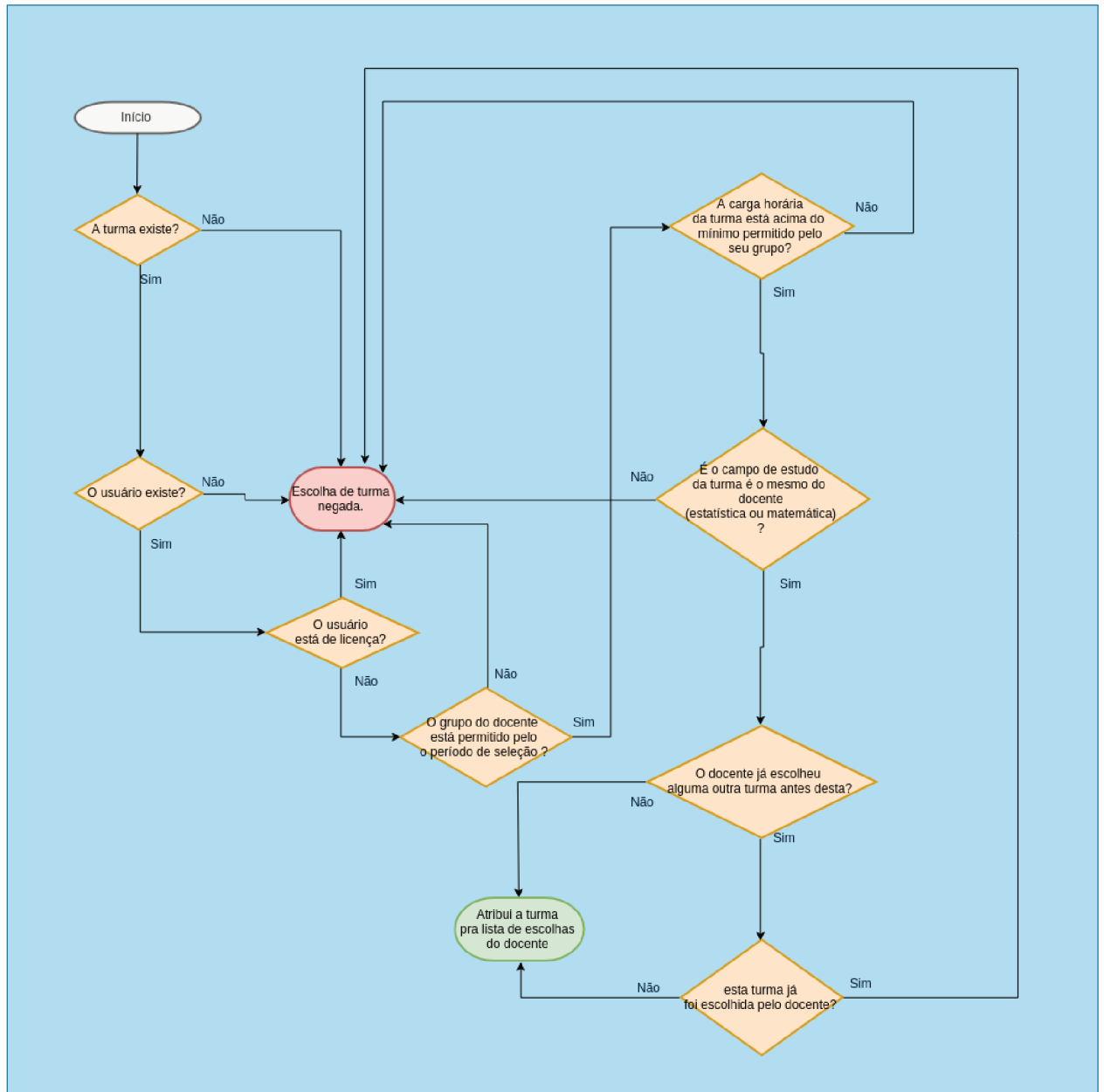


Figura 13 – Fluxo da Escolha de Turmas.

ao professor e removida da sua lista de escolhas. O sistema irá então partir para o próximo docente, se houver.

O docente pode fazer quaisquer modificações nas atribuições através da atualização dos objetos das turmas daquele ano e semestre. Podendo alterar o professor ou adicionando comentários por exemplo. Na implementação isto se dá através do ponto de acesso de atualização mostrado na listagem 3.12. Assim, o requisito funcional RF-011 foi cumprido.

Listagem 3.12 – Ponto de Acesso Para Alteração de Turmas.

```

1 router.post('/update',(req,res,next)=>{
2   if(!req.body.year){

```

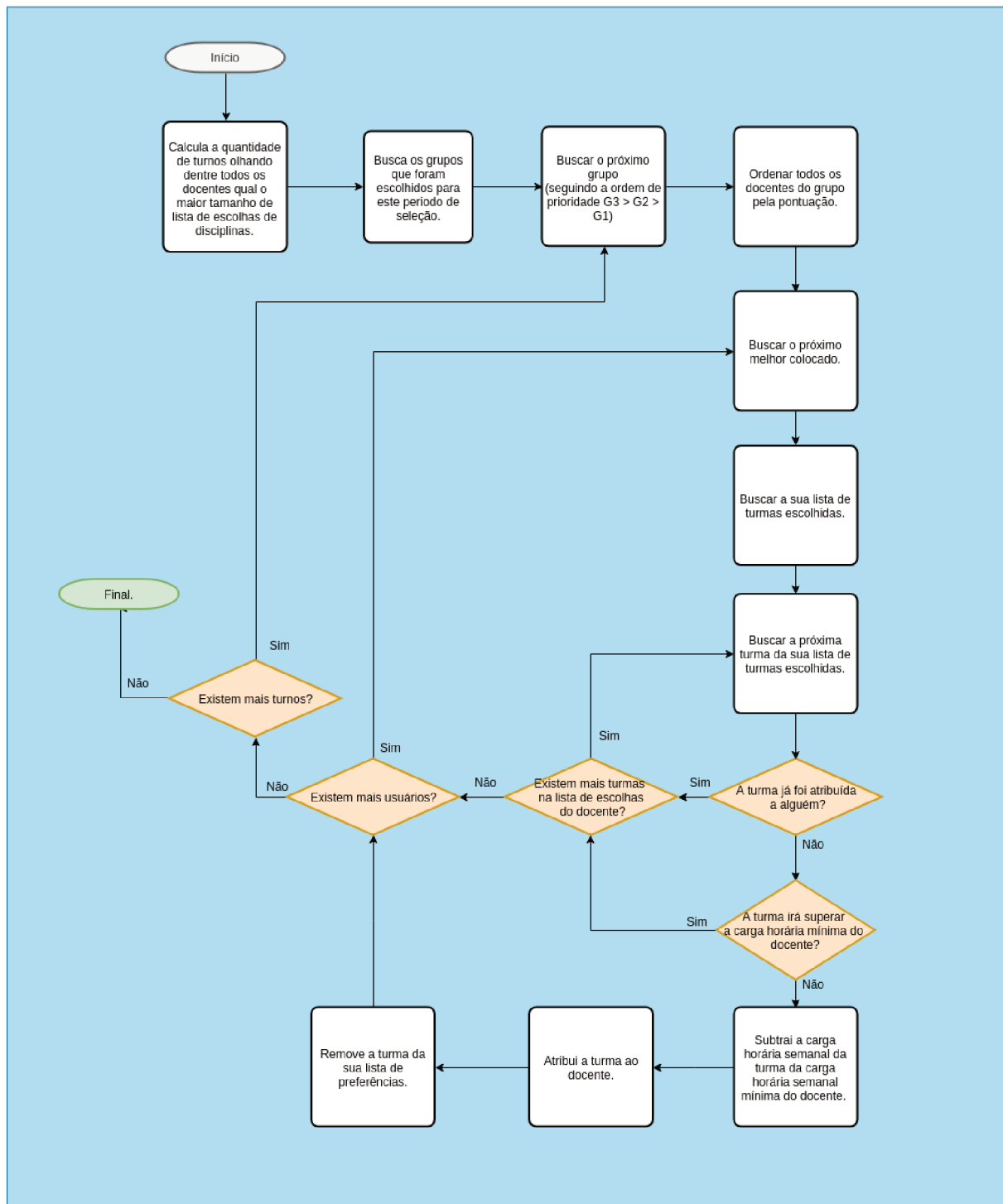


Figura 14 – Fluxo do Fechamento do Período de Seleção de Turmas: Distribuição.

```

3     res.json({success: false, msg: 'Please send the year '});
4     return;
5 }
6 var description = [];
7 if(req.body.description.length > 0){
8     description = req.body.description;
9 }
10 modifiedClass = {

```

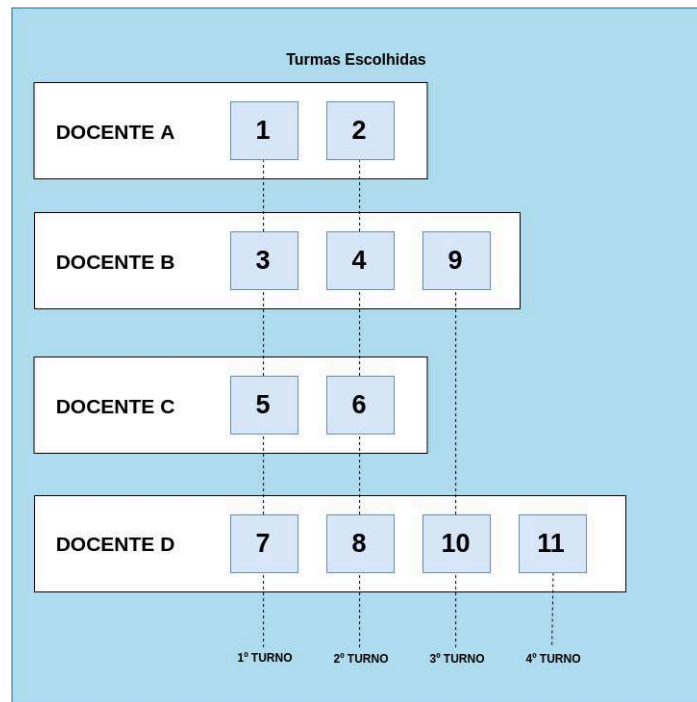


Figura 15 – Pior Caso De Distribuição.

```

11     _id          : req.body._id,
12     name         : req.body.name,
13     subjectCode  : req.body.subjectCode,
14     semester     : req.body.semester,
15     schedule     : req.body.schedule,
16     owner        : req.body.owner,
17     hours        : req.body.hours,
18     field        : req.body.field,
19     description  : description
20   }
21
22   Subjects.getSubjectByCode(modifiedClass.subjectCode, (err,
23     subject_searched) => {
24     if(err) res.json({success:false, msg: 'Failed to fetch the
25       subject :'+err});
26
27     if(subject_searched){
28       modifiedClass["field"] = subject_searched.field;
29       modifiedClass["hours"] = subject_searched.hours;
30
31       Class.updateClass(parseInt(req.body.year, 10), modifiedClass ,(
32         err, result)=>{
33         if(err){
34           res.json({success: false, msg: 'Failed to update class '+
35             err});
36         } else{

```

```
34         res.json({success: true, msg: 'Class updated!'});
35     }
36 });
37
38 }else{
39     res.json({success: false, msg: 'The classes subject doesnt
40               exist!'})
41 }
42 });
```

3.5.4.3 Revisão do Sprint

Deste sprint os requisitos foram cumpridos e funcionam. Porém ainda são necessários mais testes para a validação e encontro de possíveis falhas. E também é importante notar que do processo de distribuição algumas regras ainda precisam ser validadas pelo cliente, pois não há informação suficiente no documento fornecido. Como por exemplo a forma com que o sistema poderia atribuir disciplinas restantes para docentes que já possuem a carga horária semanal mínima.

4 Conclusão e Trabalhos Futuros

O desenvolvimento cumpre então a sua proposta em implementar uma ferramenta utilizando as tecnologias mais recentes do mercado, citadas na pilha MEAN, de forma que a ferramenta seja simples, durável, estável e unificada em uma só linguagem. Com o que fora desenvolvido o sistema já possui as funções básicas e essenciais necessárias para a manutenção e controle de docentes, atividades, turmas, tipos de atividades. E ainda realizar a distribuição de forma controlada e orientada pelo administrador do sistema.

Porém ainda que durante o desenvolvimento tenham sido realizados alguns testes básicos, o sistema ainda necessita de passar por um ambiente de homologação no qual o uso será mais próximo de uma situação real. É preciso também buscar a orientação do cliente para com as possíveis situações que não estão inclusas na resolução, mas que são importante para o uso e melhor aproveitamento da ferramenta. Algumas funcionalidades ainda devem ser implementadas, são elas:

- Adicionar funções para que as alterações em determinados objetos sejam propagadas automaticamente a todos os outros que os utilizam.
- Adicionar as devidas validações e identificadores para com as turmas do tipo sub-bloco SM ou SE tratados no artigo 12.
- Recepção de chaves *hash* para o controle de permissão aos pontos de acesso, para que apenas os usuários que fizeram login no sistema tenham capacidade de realizar as chamadas HTTP e receber as respostas.
- Alteração do tipo de validação para a função de alteração de senhas, que deverá trocar a senha recebendo e validando a senha antiga.
- Criação de um sistema de envio de emails para realização da função de “esqueci minha senha” prevista no requisito funcional RF-012.
- Geração de relatórios gerais sobre o resultado das distribuições para a análise rápida e intuitiva do administrador do sistema, possibilitando a identificação de possíveis anomalias a serem corrigidas por ele.
- Automatização da abertura e fechamento dos períodos de seleção e de registro de atividades, possibilitando com que o docente apenas configure as devidas datas e horário no sistema.

Referências

- BRAUDE, E. J.; BERNSTEIN, M. E. *Software engineering: modern approaches*. [S.l.]: Waveland Press, 2016. Citado na página 18.
- CHODOROW, K. *MongoDB: the definitive guide: powerful and scalable data storage*. [S.l.]: "O'Reilly Media, Inc.", 2013. Citado na página 16.
- CUNHA, J. P.; PEREIRA, J. L. Column-based databases: Estudo exploratório no âmbito das bases de dados nosql. In: *Atas da Conferência da Associação Portuguesa de Sistemas de Informação*. [S.l.: s.n.], 2016. v. 15, n. 15, p. 440–459. Citado na página 15.
- DIMES, T.; BARROS, R. *Scrum Essencial*. Babelcube Incorporated, 2014. ISBN 9781633397361. Disponível em: <<https://books.google.com.br/books?id=QNWZBQAAQBAJ>>. Citado na página 11.
- FALBO, R. d. A. Engenharia de software—notas de aula. *Vitória: Universidade Federal do Espírito Santo*, 2005. Citado na página 10.
- FILHO, W. de P. P. *Engenharia de software: fundamentos, métodos e padrões*. Livros Técnicos e Científicos, 2003. ISBN 9788521613398. Disponível em: <https://books.google.com.br/books?id=7c_vAAAACAAJ>. Citado na página 18.
- HAN, J. et al. Survey on NoSQL database. In: *2011 6th International Conference on Pervasive Computing and Applications*. [S.l.: s.n.], 2011. p. 363–366. Citado na página 14.
- HAVIV, A. Q. *MEAN Web Development*. [S.l.]: Packt Publishing Ltd, 2016. Citado 3 vezes nas páginas 12, 15 e 16.
- SOMMERVILLE, I. Arquitetura orientada a serviços. *Engenharia de Software*, p. 355–368, 2011. Citado 2 vezes nas páginas 10 e 11.

ANEXO A – Ciclo de distribuição de disciplina - Resolução 02/2017 - FAMAT



**SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA**



RESOLUÇÃO Nº 02/2017 DO CONSELHO DA FACULDADE DE MATEMÁTICA

Regulamenta a distribuição da carga horária didática entre os docentes da Faculdade de Matemática

O CONSELHO DA FACULDADE DE MATEMÁTICA DA UNIVERSIDADE FEDERAL DE UBERLÂNDIA, no uso de suas atribuições, em reunião realizada aos 11 dias do mês de maio do ano de 2017;

CONSIDERANDO a necessidade de revisão das normas que regulamentam a distribuição da carga horária didática da Faculdade de Matemática (FAMAT);

RESOLVE:

Art. 1º A designação da carga horária didática e das disciplinas a serem ministradas por cada docente efetivo(a) da Faculdade de Matemática no segundo semestre de cada ano letivo e no primeiro semestre do ano letivo subsequente levará em conta as atividades desenvolvidas pelo(a) docente no(s) ano(s) anterior(es), conforme consta dos Anexos 1, 2, 3 e 4 desta Resolução. Tais atividades deverão ser descritas nos Relatórios de Atividades Docente, entregues à Secretaria da FAMAT, em data preestabelecida, para que se possa:

- I. Calcular a pontuação de cada docente com base nos Anexos 1 a 4 desta Resolução.
- II. Realizar a divisão do(a)s docentes em grupos, conforme Art. 4º desta Resolução.
- III. Classificar o(a)s docentes pertencentes a cada grupo ou bloco, conforme Arts. 5º e 6º desta Resolução.
- IV. Realizar a distribuição das disciplinas, conforme sistemática descrita nos Arts. 17 e 18 desta Resolução.

Art. 2º O(A)s docentes efetivo(a)s da FAMAT serão considerados em blocos, de acordo com as áreas em que realizaram concurso e o campus universitário em que estão alocado(a)s, conforme segue:

- I. Bloco M: docentes da área da Matemática, Matemática Aplicada ou Educação Matemática alocados nos campi de Uberlândia.
- II. Bloco E: docentes da área de Estatística alocado(a)s nos campi de Uberlândia.
- III. Bloco ME: docentes da área de Matemática, Matemática Aplicada ou Educação Matemática alocados em campi externos a Uberlândia.
- IV. Bloco EE: docentes da área de Estatística alocado(a)s nos campi externos a Uberlândia.

Secretaria da Faculdade Matemática

Av. João Naves de Ávila, 2121 - Campus Santa Mônica - Bloco F – Sala 1F120
Fone (34) 3239-4156 - Fax (34) 3239-4394 – e-mail: famat@ufu.br
Uberlândia-MG - 38408-100



**SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA**



**I – DO RELATÓRIO DE ATIVIDADES DOCENTE E DA CLASSIFICAÇÃO DOS
DOCENTES SEGUNDO SUAS ATIVIDADES**

Art. 3º Os Relatórios de Atividades Docente deverão ser entregues à Secretaria da FAMAT por todo(a)s o(a)s docentes efetivos da Faculdade, no início de cada ano letivo, em período previamente especificado e em formulário próprio previamente fornecido pela Comissão de Distribuição de Aulas. A composição e as atribuições da Comissão de Distribuição de Aulas estão definidas na Seção IV desta Resolução.

§ 1º Ao ser contratado(a), cada docente será incluído(a) na última posição do grupo G3M ou G3E (dependendo da área de seu concurso). No processo de classificação subsequente, poderá optar por entregar ou não seu relatório de atividades. Se optar por entregar, será classificado(a) normalmente; caso contrário, será classificado(a) na última posição do grupo G3M ou G3E. Para efeito de terminologia, o(a)s docentes nestas condições serão nominado(a)s recém-admitido(a)s. Para efeito de desempate do(a)s recém-admitido(a)s que optarem por não entregar seus relatórios, terá classificação superior aquele(a) que tiver defendido sua tese de doutorado a menos tempo.

§ 2º O(A) docente que estiver afastado(a) para pós-graduação, estágio pós-doutoral, licença maternidade, licença saúde, licença capacitação, ou em gozo de licença prêmio por assiduidade no momento da entrega dos Relatórios de Atividades Docente, deverá apresentar seu relatório e participar de toda a sistemática prevista nesta Resolução. Neste caso, seu relatório deverá conter as atividades realizadas nos dois semestres letivos anteriores a data de início de seu afastamento, excetuando-se as atividades dos Anexos 1 e 4 que serão computadas dentro de seus prazos de validade.

§ 3º O(A) docente que estava afastado(a) em parte do período que deve constar no seu Relatório de Atividades Docente deverá apresentar seu relatório contendo as atividades realizadas nos dois semestres letivos mais recentes em que atuou por completo, excetuando-se as atividades dos Anexos 1 e 4 que serão computadas dentro de seus prazos de validade.

Art. 4º O(A)s docentes pertencentes aos blocos M e E serão preliminarmente classificado(a)s, em cada bloco, de acordo com as pontuações aferidas em seus Relatórios de Atividades Docente. Com base nesta classificação preliminar, o(a)s docentes serão subdividido(a)s em grupos. Considerando N, respectivamente P, como o número total de docentes do bloco M, respectivamente do bloco E, subtraído pelo número de docentes do mesmo bloco que estejam afastado(a)s ou em gozo de licença prêmio, com retorno programado para após o início do semestre letivo subsequente, e para os quais não há professor(a) substituto(a) contratado(a), serão constituídos os seguintes grupos:

- I. Grupo G3M: Considerando X como sendo a parte inteira do número que representa 30% (trinta por cento) de N e RM o número de docentes recém-admitido(a)s do Bloco M, este grupo será formado pelo(a)s X-RM docentes melhores classificado(a)s do Bloco M, que tenham obrigatoriamente pontuação positiva em todos os Anexos de 1 a 3, e pelos RM docentes recém-admitido(a)s do mesmo bloco. Não havendo X-RM docentes que atendam a esta exigência, o grupo G3M será completado pelo(a)s melhores classificado(a)s do Bloco M que tenham obrigatoriamente pontuação positiva em dois dos Anexos de 1 a 3.



**SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA**



- II. Grupo G3E: Considerando Y como sendo a parte inteira do número que representa 30% (trinta por cento) de P e RE o número de docentes recém-admitido(a)s do Bloco E, este grupo será formado pelo(a)s Y-RE docentes melhores classificado(a)s do Bloco E, que tenham obrigatoriamente pontuação positiva em todos os Anexos de 1 a 3, e pelo(a)s RE docentes recém-admitido(a)s do mesmo bloco. Não havendo Y-RE docentes que atendam a esta exigência, o grupo G3E será completado pelo(a)s melhores classificado(a)s do Bloco E que tenham obrigatoriamente pontuação positiva em dois dos Anexos de 1 a 3.
- III. Grupo G2M: Considerando Z como sendo a parte inteira do número que representa 30% (trinta por cento) de N, este grupo será formado pelo(a)s Z docentes melhores classificado(a)s do Bloco M, excluindo o(a)s já enquadrados no grupo G3M, que tenham obrigatoriamente pontuação positiva em pelo menos dois dos Anexos de 1 a 3.
- IV. Grupo G2E: Considerando W como sendo a parte inteira do número que representa 30% (trinta por cento) de P, este grupo será formado pelo(a)s W docentes melhores classificado(a)s do Bloco E, excluindo o(a)s já enquadrado(a)s no grupo G3E, que tenham obrigatoriamente pontuação positiva em pelo menos dois dos Anexos de 1 a 3.
- V. Grupo G1M: constituído pelo(a)s docentes do Bloco M que não foram enquadrados nos Grupos G3M e G2M.
- VI. Grupo G1E: constituído pelo(a)s docentes do Bloco E que não foram enquadrados nos Grupos G3E e G2E.

Art. 5º Após a subdivisão do(a)s docentes dos Blocos M e E em grupos, conforme disposto no Art. 4º, o(a)s docentes serão classificado(a)s internamente em cada grupo, em função das pontuações aferidas em seus Relatórios de Atividades Docente.

§ 1º Quando o grupo G3M ou G3E for completado por docentes com pontuação positiva em apenas dois dos Anexos de 1 a 3, este(a)s deverão ser classificado(a)s entre si ficando, na classificação final do respectivo grupo, abaixo do(a)s docentes do grupo que tiverem pontuação positiva em todos os Anexos de 1 a 3.

§ 2º O(A)s docentes efetivo(a)s recém-admitido(a)s que optarem por não entregar seus Relatórios de Atividades Docente serão classificado(a)s no grupo G3M ou G3E, nas últimas posições, conforme disposto no § 1º do Art. 3º.

§ 3º O(A)s docentes efetivo(a)s dos blocos M ou E que não entregarem seus Relatórios de Atividades Docente no prazo devido, excetuando-se o(a)s recém-admitidos(a) que optarem por não fazê-lo em face do disposto no § 1º do Art. 3º, serão classificado(a)s no grupo G1M ou G1E, nas últimas posições.

Art. 6º O(A)s docentes dos blocos ME e EE serão classificado(a)s em cada bloco, por campus, de acordo com as pontuações aferidas em seus Relatórios de Atividade Docente.

Parágrafo único. O(A)s docentes dos blocos referidos no *caput* que não entregarem seus Relatórios de Atividades Docente no prazo devido serão classificado(a)s nas últimas posições dos respectivos blocos.

Secretaria da Faculdade Matemática

Av. João Naves de Ávila, 2121 - Campus Santa Mônica - Bloco F – Sala 1F120
Fone (34) 3239-4156 - Fax (34) 3239-4394 – e-mail: famat@ufu.br
Uberlândia-MG - 38408-100



**SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA**



Art. 7º Para fins de desempate na classificação preliminar prevista no *caput* do Art. 4º e nas classificações internas em cada grupo ou bloco previstas nos Arts. 5º e 6º, serão observados, ordenadamente, os seguintes critérios de desempate:

- I. Maior pontuação em atividades do Anexo 1.
- II. Maior pontuação em atividades do Anexo 2.
- III. Maior pontuação em atividades do Anexo 3.
- IV. Maior tempo na carreira do magistério superior na Universidade Federal de Uberlândia.

Art. 8º Em caso de divergência entre a pontuação declarada pelo(a) docente e aquela considerada pela Comissão de Distribuição de Aulas, o(a) docente será informado(a), antes da divulgação da classificação final, e lhe será garantido um prazo de 48 horas para possíveis esclarecimentos. A pontuação e a classificação finais, determinadas conforme disposto nesta Resolução, serão informadas a cada docente antes do início da sistemática de distribuição de aulas de que trata a Seção III desta Resolução. Posteriormente, a comissão divulgará no âmbito da FAMAT as pontuações obtidas, por grupo e em ordem decrescente de classificação, sem qualquer referência ou identificação do(a) docente com a referida pontuação.

Art. 9º A divisão do(a)s docentes efetivo(a)s da FAMAT em blocos e grupos e a classificação interna do(a)s docentes em cada grupo ou bloco, conforme disposto nos Arts. 4º, 5º e 6º, serão consideradas para efeito de distribuição de aulas para os dois semestres subsequentes à realização da classificação.

Art. 10. Professore(a)s substituto(a)s ou contratado(a)s em caráter temporário não serão considerado(a)s na composição dos grupos e na classificação do(a)s docentes efetivo(a)s previstas nesta Resolução.

II – DA LISTAGEM DE DISCIPLINAS PREFERENCIAIS

Art. 11. Cada docente efetivo(a) da FAMAT, após tomar ciência de sua classificação final em um dos grupos ou blocos e do conjunto de disciplinas e horários disponíveis para o semestre letivo subsequente, deverá entregar à Secretaria da FAMAT, sempre que solicitado(a) pela Comissão de Distribuição de Aulas, uma lista, denominada Listagem de Disciplinas Preferenciais, de sugestões de disciplinas que gostaria de ministrar no semestre letivo subsequente, em ordem de preferência, na forma e data preestabelecidas pela Comissão.

Parágrafo único. Conforme previsto na sistemática de distribuição de aulas descrita no Art. 17 desta Resolução, cada docente dos blocos M e E será solicitado(a) a entregar sua Listagem de Disciplinas Preferenciais em duas ocasiões.

Art. 12. Para o fim específico da elaboração das Listagens de Disciplinas Preferenciais, as disciplinas oferecidas pela FAMAT serão agrupadas em dois blocos principais de disciplinas: um bloco constando

Secretaria da Faculdade Matemática

Av. João Naves de Ávila, 2121 - Campus Santa Mônica - Bloco F – Sala 1F120
Fone (34) 3239-4156 - Fax (34) 3239-4394 – e-mail: famat@ufu.br
Uberlândia-MG - 38408-100



**SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA**



das disciplinas das áreas de Matemática, Matemática Aplicada e Educação Matemática, que será o bloco de disciplinas sob responsabilidade do(a)s docentes do Bloco M, e outro bloco constando das disciplinas da área de Estatística, que será o bloco de disciplinas sob responsabilidade do(a)s docentes do Bloco E. Ademais, a Comissão de Distribuição de Aulas escolherá dois sub-blocos de disciplinas dentro dos blocos M e E, denominados sub-bloco SM e sub-bloco SE, respectivamente.

§ 1º A carga horária total de cada um dos sub-blocos SM e SE não poderá exceder a 15% (quinze por cento) da carga horária total do bloco principal que o contém.

§ 2º Os sub-blocos SM e SE deverão ser constituídos, entre outras, por disciplinas que apresentam dificuldade de atribuição.

Art. 13. Cada Listagem de Disciplinas Preferenciais de cada docente dos blocos M e E deverá ser constituída por, no mínimo, 15 (quinze) disciplinas relacionadas em ordem decrescente de preferência, com indicação do curso em que serão ministradas, sendo que destas pelo menos 50% devem ser de disciplinas não pertencentes aos cursos de graduação em Matemática ou Estatística da FAMAT e pelo menos duas devem ser disciplinas do turno noturno.

Art. 14. A não entrega pelo(a) docente de sua Listagem de Disciplinas Preferenciais no molde e na data solicitados, assim como a não disponibilidade das disciplinas indicadas na Listagem de Disciplinas Preferenciais do(a) docente, implicará na liberdade para a Comissão de Distribuição de Aulas atribuir para o(a) docente quaisquer disciplinas, respeitando-se os limites de carga horária específicos do grupo em que o(a) docente estiver classificado(a).

Art. 15. O(A) docente que estiver ministrando uma mesma disciplina pela segunda vez consecutiva, não poderá indicar tal disciplina em sua primeira Listagem de Disciplinas Preferenciais.

III – DO PROCESSO DE ATRIBUIÇÃO DE DISCIPLINAS

Art. 16. A atribuição de disciplinas ao(à)s docentes dos Blocos M e E deverá respeitar o limite máximo de 16 horas-aula semanais, para qualquer docente, e os seguintes limites inferiores de carga horária didática, conforme os grupos de docentes:

- I. Para docentes dos grupos G3M e G3E, mínimo de 8 horas-aula semanais;
- II. Para docentes dos grupos G2M e G2E, mínimo de 10 horas-aula semanais;
- III. Para docentes dos grupos G1M e G1E, mínimo de 12 horas-aula semanais.

§ 1º A soma da carga horária didática de cada docente efetivo(a) da FAMAT no segundo semestre de um ano com a carga horária didática do(a) mesmo(a) docente no primeiro semestre do ano letivo subsequente não poderá exceder o montante de 30 horas-aula semanais, exceto se o(a) docente autorizar carga acima desse limite.

§ 2º Ao(À) Diretor(a) da FAMAT, ao(à)s Coordenadore(a)s dos cursos de Pós-Graduação e Graduação regulares da FAMAT e ao(à) coordenador(a) do Mestrado Profissional em Ensino de Ciências e

Secretaria da Faculdade Matemática

Av. João Naves de Ávila, 2121 - Campus Santa Mônica - Bloco F – Sala 1F120
Fone (34) 3239-4156 - Fax (34) 3239-4394 – e-mail: famat@ufu.br
Uberlândia-MG - 38408-100



**SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA**



Matemática em que a FAMAT participa como unidade consorciada (caso este(a) seja docente efetivo(a) da FAMAT), que estarão em exercício da respectiva função no semestre a que se refere a distribuição de aulas, poderá ser atribuída a carga horária mínima de 6 horas-aula semanais.

§ 3º Ao(A) Assessor(a) Acadêmico(a)-Administrativo(a) que estará no exercício da função no semestre a que se refere a distribuição de aulas poderá ser atribuída a carga horária mínima de 8 horas-aulas semanais.

Art. 17. A sistemática de distribuição de aulas para o(a)s docentes dos blocos M e E será gerenciada pela Comissão de Distribuição de Aulas, que será responsável por divulgar antecipadamente o cronograma de distribuição e fazer cumprir as seguintes etapas:

- I. Os colegiados dos cursos de graduação regulares da FAMAT poderão indicar até três docentes efetivo(a)s da FAMAT, que estejam de comum acordo, para ministrar, cada um(a), uma disciplina do primeiro ou segundo períodos do respectivo curso. O(A) mesmo(a) docente não poderá ser indicado(a) mais do que duas vezes consecutivas, ainda que para disciplinas distintas.
- II. Os colegiados dos programas de pós-graduação regulares da FAMAT e do Mestrado Profissional em Ensino de Ciências e Matemática atribuirão as disciplinas dos programas para o(a)s docentes credenciado(a)s.
- III. A Comissão de Distribuição de Aulas divulgará a lista de disciplinas e horários disponíveis após a etapa II, e cada docente dos blocos M e E entregará sua primeira Listagem de Disciplinas Preferenciais que será utilizada para a atribuição de uma única disciplina a cada docente (primeira rodada de distribuição). Nesta etapa participam todo(a)s o(a)s docentes efetivo(a)s, inclusive o(a)s que já participaram das etapas anteriores e não completaram sua carga didática. A listagem de disciplinas, nesta etapa, deve obedecer aos seguintes critérios:
 - (a) Docentes dos grupos G3M ou G3E devem listar disciplinas com cargas horárias semanais maiores ou iguais a 4 horas, disciplinas dos cursos da própria FAMAT ou disciplinas pertencentes aos sub-blocos SM ou SE.
 - (b) Docentes dos grupos G2M ou G2E devem listar disciplinas com cargas horárias semanais maiores ou iguais a 5 horas, disciplinas dos cursos da própria FAMAT ou disciplinas pertencentes aos sub-blocos SM ou SE.
 - (c) Docentes dos grupos G1M ou G1E devem listar disciplinas com cargas horárias semanais maiores ou iguais a 6 horas, disciplinas dos cursos da própria FAMAT ou disciplinas pertencentes aos sub-blocos SM ou SE.Liberam-se das exigências (a)-(c) o(a)s docentes referido(a)s nos §§ 2º e 3º do Art. 16, e também aquele(a)s que já tiveram disciplinas atribuídas nas etapas I ou II da sistemática de distribuição de aulas e que necessitam de carga inferior às estabelecidas nos critérios (a)-(c) para completarem as cargas didáticas mínimas exigidas nos grupos aos quais pertencem.
- V. Com base nas Listagens de Disciplinas Preferenciais do(a)s docentes, será atribuída uma disciplina a cada docente, respeitando-se a prioridade dos grupos G3M e G3E sobre os grupos G2M e G2E, e destes sobre os grupos G1M e G1E, além da classificação interna do(a)s docentes em cada grupo. O(A)s docentes referido(a)s no § 2º do Art. 16 escolherão, nas suas



**SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA**



respectivas posições de classificação, suas cargas didáticas do semestre por completo, respeitando-se o limite mínimo estabelecido no mesmo dispositivo.

- VI. A Comissão de Distribuição de Aulas informará a cada docente a(s) disciplina(s) que lhe fora(m) atribuída(s).
- VII. A Comissão de Distribuição de Aulas divulgará a lista de disciplinas e horários disponíveis após a realização da etapa VI.
- VIII. Todo(a)s os docentes dos grupos G3M, G3E, G2M e G2E deverão entregar uma nova Listagem de Disciplinas Preferenciais, exceto o(a)s docentes referidos no § 2º do Art. 16.
- IX. Será atribuída uma disciplina a cada docente dos grupos G3M, G3E, G2M e G2E que ainda não tiver atingido a carga horária didática mínima estabelecida no Art. 16, considerando a Listagem de Disciplinas Preferenciais de cada um(a) (entregue na etapa VIII) e respeitando a classificação interna do(a)s docentes no grupo. Ao(À) docente que já tiver atingido a carga horária didática mínima poderá ou não ser atribuída uma nova disciplina, conforme necessidade ou a pedido do(a) docente.
- X. A Comissão de Distribuição de Aulas divulgará a lista de disciplinas e horários disponíveis após a realização da etapa IX.
- XI. Todo(a)s o(a)s docentes dos grupos G1M e G1E deverão entregar uma nova Listagem de Disciplinas Preferenciais, exceto o(a)s docentes listado(a)s no § 2º do Art. 16.
- XII. Nesta etapa a Comissão de Distribuição de Aulas atribuirá as disciplinas restantes. Para tal:
 - (a) Serão atribuídas disciplinas ao(à)s docentes dos grupos G1M e G1E, levando em conta as Listagens de Disciplinas Preferenciais e a classificação nos grupos, respeitando os limites de cargas horárias didáticas estabelecidos no Art. 16.
 - (b) Serão atribuídas disciplinas ao(à)s docentes dos grupos G3M, G3E, G2M e G2E que não tiverem obtido carga didática mínima, de acordo com o Art. 16, nas etapas anteriores.
- XIII. A Comissão de Distribuição de Aulas informará a cada docente as disciplinas que lhe foram atribuídas e, após respeitados os prazos e analisadas as eventuais solicitações de alteração, encaminhará a proposta de atribuição de aulas para deliberação do Conselho da FAMAT.

Parágrafo único. Nas etapas finais da sistemática de distribuição e para fins de completar o processo de distribuição de aulas, a comissão poderá, eventualmente, atribuir para alguns(mas) docentes carga horária didática menor que a mínima prevista para o seu grupo.

Art. 18. A sistemática de distribuição de aulas para o(a)s docentes dos blocos ME e EE seguirá as seguintes fases:

- I. Para cada campus, a Comissão de Distribuição de Aulas divulgará as listas de disciplinas e horários sob responsabilidade do(a)s docentes de cada bloco.
- II. Cada docente entregará sua Listagem de Disciplinas Preferenciais.
- III. A Comissão de Distribuição de Aulas fará a distribuição das disciplinas considerando a classificação do(a)s docentes de cada bloco, conforme Art. 6º, e a Listagem de Disciplinas Preferenciais de cada docente.

Secretaria da Faculdade Matemática

Av. João Naves de Ávila, 2121 - Campus Santa Mônica - Bloco F – Sala 1F120
Fone (34) 3239-4156 - Fax (34) 3239-4394 – e-mail: famat@ufu.br
Uberlândia-MG - 38408-100



**SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA**



- IV. A Comissão de Distribuição de Aulas atribuirá as disciplinas restantes ao(à)s professore(a)s substituto(a)s e contratado(a)s em caráter temporário, se houver.
- V. A Comissão de Distribuição de Aulas informará a cada docente as disciplinas que lhe foram atribuídas e, após respeitados os prazos e analisadas as eventuais solicitações de alteração, encaminhará a proposta de atribuição de aulas para deliberação do Conselho da FAMAT.

IV – DA COMISSÃO DE DISTRIBUIÇÃO DE AULAS

Art. 19. Os trabalhos referentes à análise dos Relatórios de Atividade Docente, à divisão do(a)s docentes em grupos (conforme Art. 4º), à classificação do(a)s docentes (conforme Arts. 5º e 6º) e ao gerenciamento da sistemática de distribuição de aulas, serão desenvolvidos por uma comissão, denominada Comissão de Distribuição de Aulas.

Art. 20. A Comissão de Distribuição de Aulas será nomeada pelo(a) Diretor(a) da FAMAT em até 75 (setenta e cinco) dias após o início do primeiro semestre letivo de cada ano e será composta pelos seguintes membros:

- I. O(A) Assessor(a) Acadêmico(a)-Administrativo(a) da FAMAT, como presidente;
- II. O(A)s coordenadore(a)s dos cursos de graduação regulares da FAMAT; e
- III. Dois(uas) representantes do corpo docente efetivo da FAMAT, eleito(a)s pelo Conselho da FAMAT, para a distribuição de aulas do segundo semestre letivo do ano corrente e do primeiro semestre letivo do ano subsequente.

Art. 21. Compete à Comissão de Distribuição de Aulas:

- I. Solicitar ao(à)s docentes efetivo(a)s da FAMAT seus Relatórios de Atividades Docente, nos termos estabelecidos nesta Resolução, e definir a data de entrega dos Relatórios à Secretaria da FAMAT, em tempo de se fazer cumprir esta Resolução.
- II. Solicitar ao(à)s docentes a documentação comprobatória das atividades declaradas em seus Relatórios de Atividades Docente, sempre que julgar necessário fazê-lo.
- III. Efetuar a análise dos Relatórios de Atividades Docente, classificar o(a)s docentes seguindo as pontuações estabelecidas nos Anexos 1, 2, 3 e 4, dividir o(a)s docentes em grupos, de acordo com o Art. 4º, e classificar o(a)s docentes em cada grupo ou bloco, conforme disposto nos Arts. 5º e 6º.
- IV. Dar ciência a cada docente efetivo(a) acerca de seu bloco, grupo e classificação.
- V. Elaborar e divulgar, a todos os docentes efetivos da FAMAT, o cronograma de distribuição de aulas, considerando todas as etapas previstas nos Arts. 17 e 18, além das listagens de disciplinas, com os respectivos horários, em cada etapa dos Arts. 17 e 18 em que tal procedimento está previsto.
- VI. Definir e fazer constar da solicitação da Listagem de Disciplinas Preferenciais, a forma com que cada docente efetivo(a) deverá elaborar sua Listagem, em cada etapa em que isso for necessário.

Secretaria da Faculdade Matemática

Av. João Naves de Ávila, 2121 - Campus Santa Mônica - Bloco F – Sala 1F120
Fone (34) 3239-4156 - Fax (34) 3239-4394 – e-mail: famat@ufu.br
Uberlândia-MG - 38408-100



**SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA**



- VII. Definir a cada semestre a composição dos sub-blocos SM e SE, conforme disposto no Art. 12 e em seus parágrafos.
- VIII. Cumprir a sistemática de distribuição de disciplinas estabelecida nos Arts. 17 e 18.
- IX. Dar ciência a cada docente das disciplinas que lhe foram preliminarmente designadas para o semestre subsequente e de seus respectivos horários.
- X. Deliberar, pelo voto da maioria de seus membros, acerca de solicitações encaminhadas pelo(a)s docentes após serem cientificado(a)s das disciplinas que lhes foram preliminarmente designadas e efetuar as alterações que forem consideradas pertinentes e necessárias.
- XI. Dar ciência do resultado de reexame da distribuição preliminar, conforme tal lhe seja solicitado, sempre à luz desta Resolução.
- XII. Encaminhar a proposta de distribuição da carga didática e das disciplinas ao Conselho da FAMAT para deliberação.
- XIII. Resolver os casos omissos.

Art. 22. Das decisões da Comissão de Distribuição de Aulas cabe recurso ao Conselho da FAMAT. O recurso deve ser interposto no prazo de 48 (quarenta e oito) horas após a divulgação da decisão.

V – DISPOSIÇÕES FINAIS E TRANSITÓRIAS

Art. 23. Após a apresentação da proposta de distribuição de aulas por parte da Comissão de Distribuição de Aulas e a aprovação da proposta pelo Conselho da FAMAT, apenas o surgimento de fatos novos justificarão qualquer revisão na distribuição de aulas.

§ 1º A revisão a que se refere o *caput* deste artigo será efetuada conjuntamente pelo(a) Diretor(a) e pelo(a) Assessor(a) Acadêmico(a)-Administrativo(a) da FAMAT.

§ 2º As eventuais revisões efetuadas serão informadas e justificadas, pelo(a) Diretor(a), ao Conselho da FAMAT, na primeira reunião subsequente.

Art. 24. Esta Resolução entra em vigor nesta data.

Uberlândia, 11 de maio de 2017

Marcio Colombo Fenille
Presidente

Secretaria da Faculdade Matemática

Av. João Naves de Ávila, 2121 - Campus Santa Mônica - Bloco F – Sala 1F120
Fone (34) 3239-4156 - Fax (34) 3239-4394 – e-mail: famat@ufu.br
Uberlândia-MG - 38408-100



**SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA**



ANEXOS DA RESOLUÇÃO Nº 02/2017 DO CONSELHO DA FACULDADE DE MATEMÁTICA

Anexo 1 – Atividades de pesquisa e coordenação de projetos.

- Não devem ser considerados neste anexo os artigos publicados em periódicos de divulgação científica ou iniciação científica.
- O(A) docente deverá indicar a área de avaliação de cada artigo (dentre as permitidas em cada item) e o correspondente estrato do Qualis-CAPES do periódico, observando o Qualis mais recente.
- O(A) docente deverá indicar a vigência completa de cada projeto de pesquisa, ensino ou extensão, mas, para o cálculo da pontuação correspondente, deverá considerar apenas os meses do ano anterior contidos no período de vigência, conforme Ref. 8 da tabela a seguir.
- Considerar ano civil (de 1º de janeiro a 31 de dezembro).

Ref.	ATIVIDADE	Pontuação	Pontuação máxima no período de avaliação
1	Artigo publicado em periódico científico especializado indexado no Qualis-CAPES no estrato A1 dentre as seguintes áreas de avaliação: Matemática/Probabilidade e Estatística, Educação, ou Interdisciplinar. (Válidos artigos publicados nos quatro anos anteriores)	120 pontos por artigo	-
2	Artigo publicado em periódico científico especializado indexado no Qualis-CAPES no estrato A2 dentre as seguintes áreas de avaliação: Matemática/Probabilidade e Estatística, Educação, ou Interdisciplinar. (Válidos artigos publicados nos quatro anos anteriores)	110 pontos por artigo	-
3	Artigo publicado em periódico científico especializado indexado no Qualis-CAPES no estrato B1 dentre as seguintes áreas de avaliação: Matemática/Probabilidade e Estatística, Educação, ou Interdisciplinar. (Válidos artigos publicados nos quatro anos anteriores)	100 pontos por artigo	-
4	Artigo publicado em periódico científico especializado indexado no Qualis-CAPES no estrato B2 dentre as seguintes áreas de avaliação: Matemática/Probabilidade e Estatística, Educação, ou Interdisciplinar. (Válidos artigos publicados nos quatro anos anteriores)	75 pontos por artigo	-
5	Artigo publicado em periódico científico especializado indexado no Qualis-CAPES no estrato B3 dentre as seguintes áreas de avaliação: Matemática/Probabilidade e Estatística, Educação, ou Interdisciplinar. (Válidos artigos publicados nos quatro anos anteriores)	55 pontos por artigo	-
6	Artigo publicado em periódico científico especializado indexado no Qualis-CAPES no estrato B4 dentre as	35 pontos por artigo	-

Secretaria da Faculdade Matemática

Av. João Naves de Ávila, 2121 - Campus Santa Mônica - Bloco F – Sala 1F120
Fone (34) 3239-4156 - Fax (34) 3239-4394 – e-mail: famat@ufu.br
Uberlândia-MG - 38408-100



**SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA**



	seguintes áreas de avaliação: Matemática/Probabilidade e Estatística, Educação, ou Interdisciplinar. (Válidos artigos publicados nos quatro anos anteriores)		
7	Artigo publicado em periódico científico que não pertença ao Qualis-CAPES das áreas Matemática/Probabilidade e Estatística, Educação, ou Interdisciplinar, em nenhuma das quatro últimas avaliações do Qualis-CAPES, mas que esteja em algum dos estratos A1, A2, B1, B2, B3 ou B4 de outra área do Qualis-CAPES. (Válidos artigos publicados nos quatro anos anteriores)	10 pontos por artigo	-
8	Coordenação de projeto de pesquisa, ensino ou extensão, com financiamento externo, sem remuneração complementar e/ou bolsa para o (a) coordenador(a), comprovada por documento de aprovação do projeto pelo órgão de fomento	5 pontos por mês de coordenação no ano anterior	-

Anexo 2 – Atividades de capacitação, orientação, extensão e administração.

- Devem ser consideradas as atividades realizadas no ano civil (de 1º de janeiro a 31 de dezembro) anterior à entrega do Relatório de Atividades Docente.

Ref.	ATIVIDADE	Pontuação	Pontuação máxima no período de avaliação
1	Tese de doutorado defendida pelo(a) docente	120	120
2	Orientação de tese de doutorado	15 pontos por aluno/mês completo	540
3	Orientação de dissertação de mestrado	10 pontos por aluno/mês completo	360
4	Orientação de PIVIC, PIBIC ou projetos de iniciação científica oriundos de projetos de pesquisa com financiamento externo	5 pontos por aluno/mês completo	180
5	Orientação de bolsista do PET (Programa de Educação Tutorial) em pesquisa individual e/ou coletiva	3 pontos por mês completo	72
6	Outras orientações de iniciação científica aprovadas por órgãos oficiais	3 pontos por aluno/mês completo	72

Secretaria da Faculdade Matemática

Av. João Naves de Ávila, 2121 - Campus Santa Mônica - Bloco F – Sala 1F120
Fone (34) 3239-4156 - Fax (34) 3239-4394 – e-mail: famat@ufu.br
Uberlândia-MG - 38408-100



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA



7	Orientação de monografia de cursos de especialização não remunerados oferecidos pela FAMAT	4 pontos por aluno/mês completo	96
8	Orientação de Trabalhos de Conclusão de Curso nos cursos de graduação (Não computar orientações já consideradas nos itens 4, 5 e 6)	2 pontos por aluno/mês completo	48
9	Membro titular de banca de defesa de doutorado	30	120
10	Membro titular de banca de defesa de mestrado	15	60
11	Membro titular de banca de defesa de monografia de pós-graduação <i>lato sensu</i>	5	20
12	Membro titular de banca de defesa de monografia de graduação	4	16
13	Membro titular de banca de qualificação de doutorado	10	40
14	Membro titular de banca de qualificação de mestrado	5	20
15	Exercício da Diretoria da FAMAT	30 pontos por mês completo	-
16	Exercício de direção no âmbito da UFU. (Item incluído pela Resolução nº 02/2014 do CONFAMAT, de 04/12/2014)	10 pontos por mês completo	-
17	Exercício de Coordenação de curso de graduação ou programa de pós-graduação regular da FAMAT ou programa de pós-graduação consorciado	15 pontos por mês completo	-
18	Exercício de Assessoria Acadêmica-Administrativa da FAMAT	10 pontos por mês completo	-
19	Coordenação da Semana da Matemática/Estatística ou de eventos científicos promovidos pela FAMAT	60 pontos por evento	60
20	Coordenação de eventos científicos não pontuados no item 19	30 pontos por evento	30
21	Membro da Comissão Organizadora da Semana da Matemática/Estatística da FAMAT ou de eventos científicos promovidos pela FAMAT	15 pontos por evento	30
22	Membro da Comissão Organizadora de eventos científicos não pontuados no item 21	10 pontos por evento	20
23	Membro de Comitê Científico de eventos	5 pontos por evento	10
24	Presidência do Comitê Editorial da Revista Eletrônica Matemática e Estatística em Foco ou de outra revista	40 pontos por edição	-

Secretaria da Faculdade Matemática

Av. João Naves de Ávila, 2121 - Campus Santa Mônica - Bloco F – Sala 1F120
Fone (34) 3239-4156 - Fax (34) 3239-4394 – e-mail: famat@ufu.br
Uberlândia-MG - 38408-100



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA



	científica da área de Matemática, Matemática Aplicada, Educação Matemática ou Estatística	publicada	
25	Membro do Comitê Editorial da Revista Eletrônica Matemática e Estatística em Foco ou de outra revista científica	5 pontos por edição publicada	-
26	Coordenação de núcleo, do PICME ou laboratório da FAMAT	2 pontos por mês completo	-
27	Presidência de Núcleo Docente Estruturante de curso de graduação da FAMAT	3 pontos por mês completo	-
28	Presidência da Câmara de Pesquisa, Extensão e Pós-Graduação da FAMAT	3 pontos por mês completo	-
29	Coordenação de estágio supervisionado	5 pontos por mês completo	-
30	Membro do Conselho da FAMAT	5 pontos por mês completo	-
31	Membro de colegiado de curso ou Núcleo Docente Estruturante de curso da UFU (O presidente do Núcleo Docente Estruturante pontua somente na Ref. 27 acima)	2 pontos por mês completo	-
32	Representação da FAMAT junto à Biblioteca	1 ponto por mês completo	-
33	Representação da FAMAT junto ao Conselho de Extensão, Cultura e Assuntos Estudantis	1 ponto por mês completo	-
34	Membro da Comissão Interna de Avaliação de Desempenho Docente (CIADD)	5 pontos por mês completo	-
35	Membro da Comissão de Distribuição de Aulas	6 pontos por distribuição	-
36	Membros de comissões internas da Unidade Acadêmica	3 pontos por comissão	-
37	Membro de comissão ou comitê da UFU externo a FAMAT	2 pontos por comissão/ mês completo	48
38	Membro de banca de concurso público para professor(a) efetivo(a)	15	-
39	Membro de banca de processo seletivo para contratação de professore(a)s substituto(a)s	8	-
40	Parecer <i>ad-hoc</i> para publicações em periódicos com corpo editorial	5	20

Secretaria da Faculdade Matemática

Av. João Naves de Ávila, 2121 - Campus Santa Mônica - Bloco F – Sala 1F120
Fone (34) 3239-4156 - Fax (34) 3239-4394 – e-mail: famat@ufu.br
Uberlândia-MG - 38408-100



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA



41	Parecer <i>ad-hoc</i> para órgãos oficiais de utilidade pública	2	20
42	Coordenação de projeto de extensão sem remuneração complementar e/ou bolsa para o(a) coordenador(a), sem financiamento e aprovado pelo Conselho da FAMAT. (Válidos somente projetos com pelo menos 30 horas)	30	60
43	Coordenação de projeto de extensão sem remuneração complementar e/ou bolsa para o(a) coordenador(a), com financiamento e aprovado pelo Conselho da FAMAT. (Válidos somente projetos com pelo menos 30 horas)	50	100

Anexo 3 – Atividades de ensino (sem remuneração complementar).

- Devem ser consideradas as atividades de ensino realizadas nos dois semestres do ano letivo anterior à entrega do Relatório de Atividades Docente.

Ref.	ATIVIDADE	Pontuação
1	Disciplina ministrada em cursos de graduação, atribuída ao(à) docente pela FAMAT, com aprovação do Conselho da Unidade.	Multiplicar o número de horas-aula semanais da disciplina por 10 (dez). Para disciplina ministrada para turma com mais de 40 estudantes, adicionar 0,25 ponto por estudante acima de 40. (O número de estudantes da turma deve ser entendido como aquele constante do Registro de Resultados da disciplina gerado pelo Portal do Docente).
2	Disciplina ministrada em cursos de pós-graduação <i>stricto sensu</i> cadastradas na FAMAT.	Multiplicar o número de horas-aula semanais da disciplina por 10 (dez).
3	Coordenação de projeto de ensino sem remuneração complementar e/ou bolsa para o(a) coordenador(a), aprovado pelo Conselho da FAMAT. (Válidos somente projetos com pelo menos 30 horas, limitado a dois projetos).	30 pontos por projeto

Secretaria da Faculdade Matemática

Av. João Naves de Ávila, 2121 - Campus Santa Mônica - Bloco F – Sala 1F120
Fone (34) 3239-4156 - Fax (34) 3239-4394 – e-mail: famat@ufu.br
Uberlândia-MG - 38408-100



**SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA**



Anexo 4 – Outras atividades.

- Considerar ano civil (de 1º de janeiro a 31 de dezembro).

Ref.	ATIVIDADE	Pontuação	Pontuação máxima no período de avaliação
1	Publicação de livro com selo de editora com corpo editorial. (Válidos livros publicados nos dois anos anteriores)	120	120
2	Publicação de capítulo de livro com selo de editora com corpo editorial. (Válidos livros publicados nos dois anos anteriores)	30	60
3	Palestra/conferência/apresentação oral de trabalho em evento científico internacional. (Válidas somente apresentações realizadas pelo(a) próprio(a) docente no ano anterior)	20	60
4	Minicurso/oficina em evento científico internacional. (Válidas somente apresentações realizadas pelo(a) próprio(a) docente no ano anterior)	30	60
5	Palestra/conferência/apresentação oral de trabalho em evento científico nacional. (Válidas somente apresentações realizadas pelo(a) próprio(a) docente no ano anterior)	10	30
6	Minicurso/oficina ministrado(a) em evento científico nacional. (Válidas somente apresentações realizadas pelo(a) próprio(a) docente no ano anterior).	15	30
7	Apresentação de pôster de trabalho em reunião científica. (Válidas somente apresentações realizadas pelo(a) próprio(a) docente no ano anterior)	5	15
8	Artigo publicado em periódico científico especializado, com corpo editorial. (Válidos artigos publicados no ano anterior, não pontuados no Anexo 1).	10	20

Secretaria da Faculdade Matemática

Av. João Naves de Ávila, 2121 - Campus Santa Mônica - Bloco F – Sala 1F120
Fone (34) 3239-4156 - Fax (34) 3239-4394 – e-mail: famat@ufu.br
Uberlândia-MG - 38408-100