

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

João Augusto Locatelli

Sistema Online para Distribuição de Disciplinas

Uberlândia, Brasil

2015

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

João Augusto Locatelli

Sistema Online para Distribuição de Disciplinas

Trabalho de Conclusão de Curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Dr. Bruno Augusto Nassif Travençolo

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2015

João Augusto Locatelli

Sistema Online para Distribuição de Disciplinas

Trabalho de Conclusão de Curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Trabalho aprovado.

Uberlândia, Brasil, 11 de dezembro de 2015:

Dr. Bruno Augusto Nassif Travençolo
Orientador

Prof. Dr. rer. nat. Daniel Duarte
Abdala

Dr. André Ricardo Backes

Uberlândia, Brasil
2015

Resumo

Na Faculdade de Computação da Universidade Federal de Uberlândia o processo de distribuição de disciplinas acontece todo semestre, atendendo o desejo dos professores na seleção das disciplinas preferenciais a serem ministradas, mas respeitando um conjunto de regras. Atualmente esse processo é realizado manualmente por uma comissão responsável, demandando muito tempo e esforço. Sendo assim, o objetivo deste trabalho é o desenvolvimento de um software *web* que auxilie os professores no processo de seleção e classificação de prioridades de disciplinas e também ajude a comissão de distribuição na definição das atribuições das disciplinas. Espera-se com a criação deste sistema que o processo de distribuição de disciplinas seja facilitado e que seja disponibilizada uma interface mais amigável aos professores para realizarem o cadastro e consulta de informações em uma base de dados.

Palavras-chave: Distribuição de disciplinas, Sistema online, Universidade Federal de Uberlândia, Faculdade de computação.

Lista de Ilustrações

Figura 1 – Interação entre as camadas em projeto utilizando o padrão arquitetural Model-View-Controller (W3SCHOOLS, 2015).	11
Figura 2 – Grupos de processos de gerenciamento de projetos (INSTITUTE, 2012).	13
Figura 3 – Os grupos de processos interagem em uma fase ou em um projeto (INSTITUTE, 2012).	13
Figura 4 – Grupo de processos de planejamento (INSTITUTE, 2012).	15
Figura 5 – Planilha utilizada para distribuição de disciplinas.	19
Figura 6 – Diagrama de casos de uso dos requisitos do sistema.	22
Figura 7 – Diagrama entidade-relacionamento do modelo de dados criado para o sistema.	23
Figura 8 – Organização das classes presentes no software.	24
Figura 9 – Diagrama de classes do software.	25
Figura 10 – Alteração da fila de disciplinas do docente.	27
Figura 11 – Consultas para avaliar filas de disciplinas e professores.	28
Figura 12 – Layout utilizado para modificar a prioridade das turmas na lista.	29
Figura 13 – Alteração de prioridade utilizando a funcionalidade “arrastar e soltar”.	29
Figura 14 – Funcionalidade para visualizar e comparar horários entre turmas (as turmas que não possuem conflitos em seus horários aparecem em azul e as que possuem aparecem em vermelho).	30
Figura 15 – Sistema apenas com os menus para realizar consultas de informações.	30

Lista de Tabelas

Tabela 1 – Ferramentas utilizadas.	18
--	----

Lista de abreviaturas e siglas

CDD	Comissão de Distribuição de Disciplinas
CSS	<i>Cascade Style Sheets</i>
FACOM	Faculdade de Computação
MVC	Modelo – Visão – Controle
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
JS	<i>JavaScript</i>
JSON	<i>JavaScript Object Notation</i>
REST	<i>REpresentational State Transfer</i>
SQL	<i>Structured Query Language</i> ou Linguagem de Consulta Estruturada
URL	<i>Universal Resource Locator</i>

Sumário

1	INTRODUÇÃO	8
2	OBJETIVOS	10
2.1	Objetivo Geral	10
2.2	Objetivos Específicos	10
3	REFERENCIAL TEÓRICO	11
3.1	Arquitetura	11
3.1.1	Padrão Arquitetural MVC (<i>Model – View - Controller</i>)	11
3.2	Gerenciamento de projetos	12
3.2.1	Grupo de processos de iniciação	13
3.2.2	Grupos de processos de planejamento	14
3.2.3	Grupo de processos de execução	15
3.2.4	Grupo de processos de monitoramento e controle	16
3.2.5	Grupo de processos de encerramento	16
4	MATERIAIS E MÉTODOS	17
4.1	Materiais	17
4.2	Fases do projeto	18
4.2.1	Iniciação	18
4.2.2	Planejamento	19
4.2.3	Execução	22
4.2.3.1	Construção do modelo de dados	23
4.2.3.2	Desenvolvimento dos serviços (<i>“Back end”</i>)	23
4.2.3.3	Desenvolvimento da interface web (<i>“Front end”</i>)	25
4.2.4	Encerramento	26
5	RESULTADOS OBTIDOS	27
6	CONSIDERAÇÕES FINAIS	31
	REFERÊNCIAS	32

1 Introdução

O processo de distribuição de disciplinas da Faculdade de Computação (FACOM) da Universidade Federal de Uberlândia é uma atividade realizada todo semestre por uma comissão chamada Comissão de Distribuição de Disciplinas (CDD). O trabalho da CDD é definir quais disciplinas serão ofertadas no próximo semestre e quais professores as ministrarão. O trabalho da comissão é relativamente complexo, uma vez que são ofertadas por volta de 150 turmas de disciplinas, totalizando aproximadamente 600 horas-aula. Essas turmas são distribuídas entre aproximadamente 70 professores, sendo que a carga horária atribuída a cada professor é distinta, pois depende de fatores como o regime de trabalho do docente (e.g., Dedicção Exclusiva, 20h semanais, professor substituto) e cargos administrativos ou acadêmicos.

Esse processo é muito importante, pois a cada distribuição é buscado atender o desejo dos professores em ministrar suas disciplinas com maior preferência para o próximo semestre, respeitando a carga horária existente e respeitando as regras definidas no Conselho da Faculdade. As disciplinas existentes na FACOM são fixas, sendo que para cada semestre são criadas turmas para essas disciplinas, podendo ocorrer a criação de uma ou mais turmas para a mesma disciplina. Um dos fatores que definem o número de turmas que serão criadas é a quantidade de alunos que necessitam cursar uma disciplina, por exemplo. A situação ideal ao final de todo o processo é atingir a carga horária pré-definida de cada professor e distribuir um professor para cada turma existente, evitando que a turma fique sem um docente responsável.

A forma como é realizada a distribuição dos docentes nas turmas criadas para o semestre respeita uma fila de professores. Cada disciplina possui uma fila, formada por docentes que manifestaram interesse em ministrar a disciplina. O professor que está no topo da fila, será a pessoa que com maior chance de ministrar a turma da respectiva disciplina no próximo semestre. A primeira vez em que foram criadas as filas, elas foram ordenadas pela data de ingresso do docente na FACOM e foi dado ao docente a oportunidade de entrar em qualquer fila.

Antes do início da distribuição, a CDD faz um chamado aos docentes para que estes definam suas prioridades – isto é, dentro das disciplinas em que o docente esta na fila, ele deve definir uma prioridade (ou ordem de preferência) para as disciplinas que ele deseja ministrar. A CDD vai então tentar atribuir as disciplinas ao docente seguindo a ordem de prioridade definida por ele.

Quando o professor no topo da fila não pode ministrar determinada turma por causa de algum impedimento, o próximo docente da fila é analisado, e assim sucessi-

vamente até a turma possuir um responsável. Os impedimentos podem ocorrer de três maneiras: (i) o professor possui algum impedimento (alguma outra atividade realizada na faculdade) no horário exigido para ministrar a turma; (ii) a carga horária do professor já foi excedida e não pode lhe ser atribuído outras turmas; (iii) existe outra turma (com uma maior prioridade) que já foi atribuída para esse professor e seu horário entra em conflito com o horário da turma sendo analisada.

Até o semestre 2015-02 todo esse processo era realizado manualmente pela CDD, demandando muito tempo, esforço e tornando o processo propenso a erros. O cadastro das informações das preferências de cada professor era realizado em uma planilha compartilhada, em que o docente indica uma lista de turmas que deseja ministrar e sua respectiva prioridade. A CDD utilizava essas informações em conjunto com os horários definidos das turmas para realizar uma distribuição manual. Esse método apresenta uma série de dificuldades devido ao grande número de combinações existentes entre turmas e professores, conflitos de horários, conflitos de preferências de diferentes professores, entre outros.

Considerando os problemas apresentados, percebeu-se a necessidade de automatizar todo o processo de distribuição de disciplinas e facilitar aos professores o cadastro e visualização das informações de filas e prioridades existentes. Dessa forma, o objetivo deste trabalho é a criação de um sistema *Web* para auxiliar em todo o processo de distribuição, de forma a facilitar tanto à CDD quanto aos professores da FACOM o acesso às informações sobre as disciplinas, seus horários, turmas e informações sobre preferências dos professores em ministrar disciplinas. O sistema foi criado e já foi possível utilizá-lo para a distribuição do semestre de 2016-01

2 Objetivos

2.1 Objetivo Geral

Construção de um sistema web para facilitar o processo de distribuição de disciplinas que ocorre na Faculdade de Computação.

2.2 Objetivos Específicos

A partir do objetivo geral, podemos enumerar os seguintes objetivos específicos com as características desejadas no sistema:

- Fácil usabilidade para os professores por meio de uma interface web acessível via browser.
- Criação de um modelo de dados lógico das informações em um banco de dados relacional.
- Visualizar histórico de informações de diferentes semestres.
- Auxiliar a CDD no gerenciamento de informações de filas e preferências de docentes.
- Permitir que os processos de entrada e saída de disciplinas e definição de prioridades nas turmas sejam realizados em etapas diferentes.
- Facilitar o acesso à informação de filas de disciplinas, turmas e professores.

3 Referencial Teórico

Será apresentado neste Capítulo todo o estudo realizado para o desenvolvimento do projeto.

3.1 Arquitetura

A complexidade na construção de softwares vem aumentando, exigindo maior tempo no desenvolvimento e gerando uma maior dificuldade para realizar mudanças e manutenções necessárias no desenvolvimento contínuo. Para diminuir esses problemas é fundamental a definição de um padrão de arquitetura a ser utilizado, de forma que atenda as necessidades do software sendo desenvolvido.

A arquitetura estabelece uma relação e interação entre componentes gerando um impacto nos custos, tempo, desempenho e robustez no desenvolvimento de softwares.

3.1.1 Padrão Arquitetural MVC (*Model – View - Controller*)

O padrão arquitetural *Model-View-Controller* permite a separação do sistema em três camadas lógicas ([Figura 1](#)) com funções distintas com as responsabilidades de exibir informações e interagir com o usuário, capturar dados de entradas e retornar respostas apropriadas e representar o domínio do problema e seu estado, bem como métodos para acesso e alterações quando necessário.

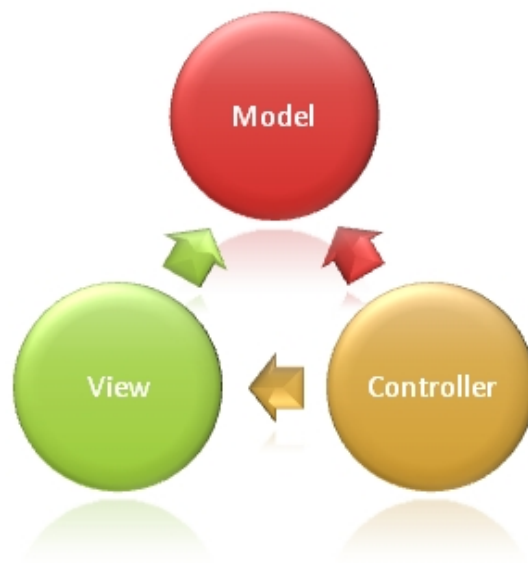


Figura 1 – Interação entre as camadas em projeto utilizando o padrão arquitetural Model-View-Controller ([W3SCHOOLS, 2015](#)).

A camada de modelo (*Model*) é responsável pelo controle lógico dos dados da aplicação e seus estados. A camada de visão (*View*) possui a tarefa de interagir e exibir para o usuário as informações do sistema. E por último a camada de controle (*Controller*) é responsável por lidar com a interação do usuário lendo os dados de entradas da camada de visão, enviar essas informações para a camada de modelo e retornar uma resposta.

3.2 Gerenciamento de projetos

O gerenciamento de projetos é a aplicação de conhecimento, habilidades ferramentas e técnicas às atividades do projeto a fim de atender seus requisitos. O gerenciamento de projetos é realizado por meio da aplicação e integração apropriadas dos 42 processos agrupados logicamente abrangendo os 5 grupos (INSTITUTE, 2012). Os cinco grupos são:

- Iniciação
- Planejamento
- Execução
- Monitoramento e controle
- Encerramento

Os processos de gerenciamento de software são utilizados como guia para auxiliar no desenvolvimento de um projeto. Os grupos de processos de gerenciamento de projetos são vinculados pelas saídas que produzem. Raramente os grupos de processos são eventos distintos ou que ocorrem uma única vez; são atividades sobrepostas que ocorrem ao longo de todo o projeto. A saída de um processo em geral torna-se uma entrada em outro processo ou é uma entrega do projeto (Figura 2) (INSTITUTE, 2012).

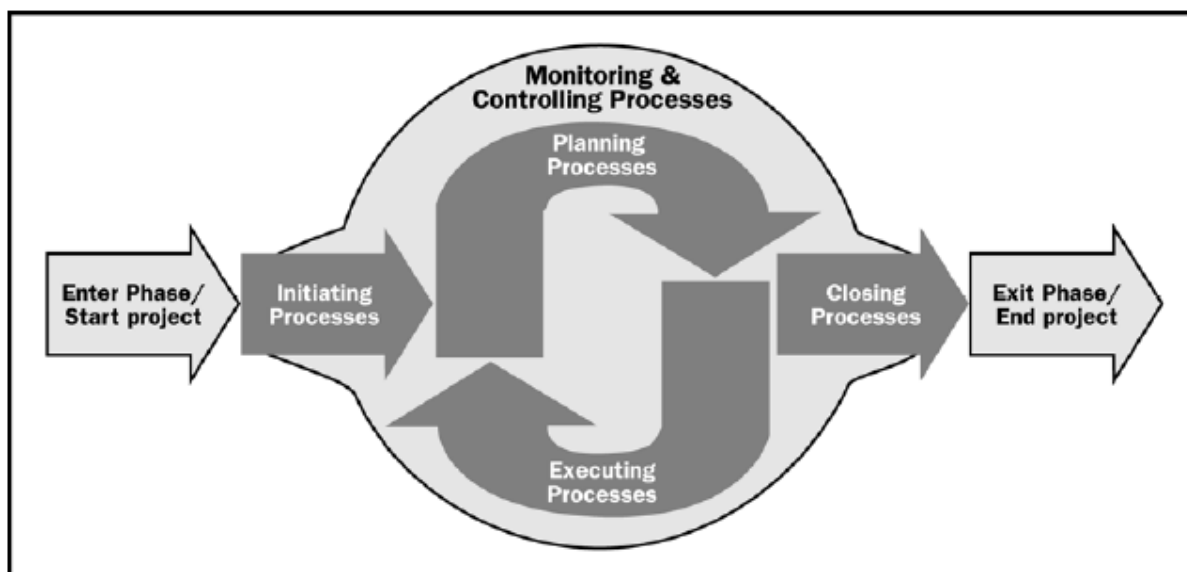


Figura 2 – Grupos de processos de gerenciamento de projetos (INSTITUTE, 2012).

A Figura 3 exemplifica como os diferentes grupos de processos interagem entre si durante o período completo de desenvolvimento de um projeto.

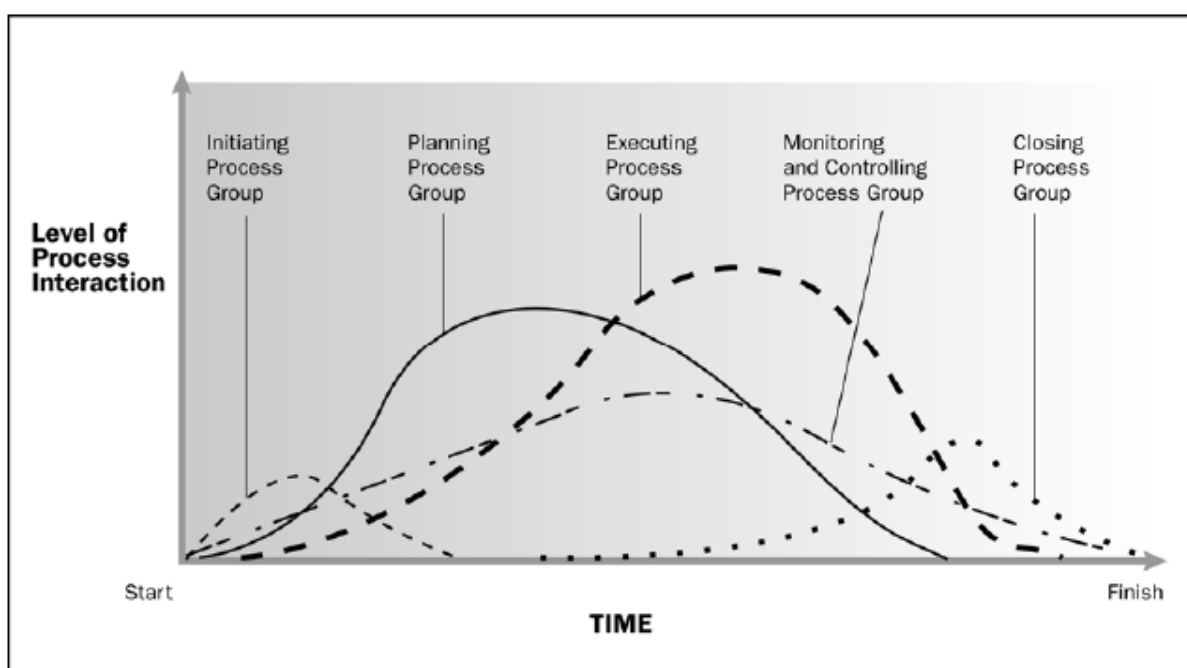


Figura 3 – Os grupos de processos interagem em uma fase ou em um projeto (INSTITUTE, 2012).

3.2.1 Grupo de processos de iniciação

No grupo de processos de iniciação o escopo inicial do projeto é definido e seus recursos financeiros são alocados. Durante essa etapa ocorre o desenvolvimento de um

documento para formalizar a autorização de desenvolvimento do projeto e documentar os requisitos iniciais que satisfaçam as partes interessadas.

3.2.2 Grupos de processos de planejamento

Durante o grupo de processos de planejamento todos os esforços são focados em definir o escopo total do projeto e detalhar seus requisitos. Nesta etapa o plano de execução é construído para ser seguido na próxima fase. Os processos de planejamento desenvolvem o plano de gerenciamento e os documentos do projeto que serão usados para executá-lo. A natureza multidimensional do gerenciamento de projetos cria loops de feedback periódicos para análise adicional. À medida que mais informações ou características do projeto são coletadas e entendidas, pode ser necessário um planejamento adicional ([INSTITUTE, 2012](#)). A [Figura 4](#) exibe todos os processos necessários para o planejamento de um projeto e suas interações existentes.

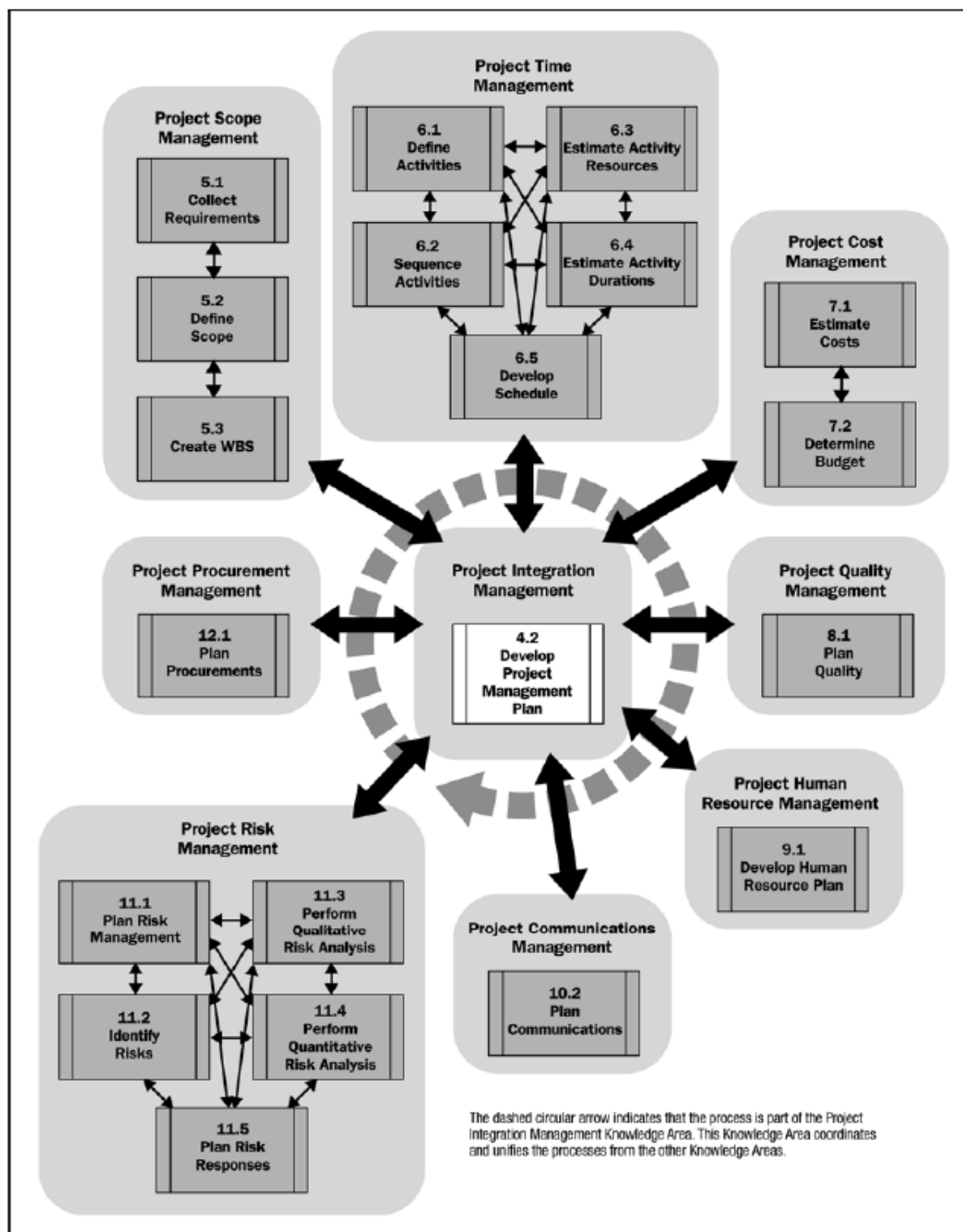


Figura 4 – Grupo de processos de planejamento (INSTITUTE, 2012).

3.2.3 Grupo de processos de execução

O grupo de processos de execução segue o plano definido durante a etapa de planejamento para cumprir as especificações do projeto e concluir o trabalho definido.

Algumas das atividades realizadas nesta etapa incluem orientar e gerenciar a execução do projeto, realizar a garantia da qualidade, gerenciar a equipe do projeto e gerenciar as expectativas das partes interessadas.

3.2.4 Grupo de processos de monitoramento e controle

O grupo de Processos de Monitoramento e Controle consiste nos processos necessários para acompanhar, revisar e regular o progresso e o desenvolvimento do projeto, identificar todas as áreas nas quais serão necessárias mudanças no plano e iniciar as mudanças correspondentes. O principal benefício deste grupo de processos é que o desempenho do projeto é observado e mensurado de forma periódica e uniforme para identificar variações em relação ao plano de gerenciamento do mesmo ([INSTITUTE, 2012](#)).

3.2.5 Grupo de processos de encerramento

O grupo de processos de encerramento consiste no termino formal do projeto, finalizando todas as atividades pendentes de todos os grupos de processos anteriores. Nesta etapa busca-se obter a aceitação do cliente, realizar uma revisão pós-projeto e documentar lições aprendidas.

4 Materiais e Métodos

Nesta seção serão apresentados o ambiente e ferramentas utilizadas para o desenvolvimento do software e serão explicados todos os passos envolvidos no projeto, desde seu início até a sua conclusão.

4.1 Materiais

A Tabela 1 lista todas as ferramentas utilizadas para o desenvolvimento do projeto:

Ferramenta	Versão	Descrição
Computador para desenvolvimento	-	Especificações: processador Intel(R) Core(TM) i5-4690 CPU 3.50 GHz, placa de vídeo AMD Radeon HD 7870, 8GB de memória RAM, 500GB de espaço em disco rígido e sistema operacional Windows 8.1 Pro 64-bit.
Computador como servidor	-	Especificações: processador Intel(R) Core(TM)2 Quad CPU Q8400 2.66GHz, 4GB de memória RAM, 300GB de espaço em disco rígido e sistema operacional Windows 7 Profissional 32-bit.
Java	1.7.0-79	Linguagem de programação utilizada para a construção do sistema para a plataforma <i>web</i> .
Apache Tomcat	8.0.26	O Apache Tomcat é um servidor de aplicações Java de código aberto. Ele foi utilizado devido a sua fácil instalação e rápida iniciação.
Eclipse	Mars Release (4.5.0)	O Eclipse foi escolhido para o ambiente de desenvolvimento por possuir uma variedade de benefícios, como o recurso de <i>autocomplete</i> que permite acessar o nome de classes e variáveis criadas em todo o projeto, integração com o servidor de aplicação Tomcat possibilitando um rápido <i>deploy</i> (instalação da aplicação no servidor), controle sobre o servidor, ambiente próprio para realizar o <i>debug</i> das funcionalidades do sistema, entre outros.

Hibernate	4.2.15.Final	O Hibernate é um <i>framework</i> que foi utilizado para auxiliar o desenvolvimento do projeto utilizando um banco de dados no modelo relacional em conjunto com a linguagem Java. Com a utilização dessa ferramenta é possível realizar o mapeamento das entidades presente no banco de dados em classes orientadas a objetos. Ele também foi utilizado para auxiliar na realização de consultas e persistência de informações diminuindo a complexidade dessas operações.
Apache Maven	3.3.3	Foi utilizado para gerenciar as dependências do projeto. Por meio desta ferramenta é possível automatizar o <i>download</i> de <i>frameworks</i> requeridos, organizar todas as dependências em um único local e verificar/atualizar suas versões.
RESTEasy	3.0.4.Final	É um <i>framework</i> que auxilia na construção de projetos Java RESTful. REST (<i>REpresentational State Transfer</i>) é um estilo de arquitetura para sistemas de hipermídia distribuídos, como a <i>World Wide Web</i> . O conceito dos recursos identificados por identificadores de recursos universais (URIs) é central para a arquitetura RESTful. Esses recursos podem ser manipulados usando uma interface padrão, como HTTP, e as informações são trocadas usando representações desses recursos (NETBEANS, 2015). Utilizando esse <i>framework</i> foi possível realizar a comunicação entre as páginas HTML com os serviços criados em Java por meio do protocolo HTTP.
PostgreSQL	9.4.5	É o sistema gerenciador de banco de dados relacional utilizado no projeto.
Google-gson	2.2.2	Biblioteca Java utilizada para converter classes do projeto em uma <i>string</i> JSON e vice-versa.
Bootstrap	v3.3.5	Bootstrap é o <i>framework HTML</i> , CSS e JS mais popular para o desenvolvimento web (BOOTSTRAP, 2015) e foi utilizado para a criação do layout do projeto.
AngularJS	v1.4.6	AngularJS é um <i>framework</i> estrutural para o desenvolvimento de projetos web dinâmicos (ANGULARJS, 2015).

Tabela 1 – Ferramentas utilizadas.

4.2 Fases do projeto

4.2.1 Iniciação

A fase de iniciação foi onde ocorreu o levantamento do escopo do projeto e suas

principais necessidades. Durante essa primeira etapa foi definida uma data limite para o termino do projeto, dia 12/11/2015, pois nesta data se iniciaria todo o processo de distribuição de disciplinas para o semestre 2016-01. Nesse período foi acordado que o software seria desenvolvido para a plataforma *web* sem haver a necessidade de instalação ou configuração no computador do usuário e a definição da linguagem Java a ser utilizada para o desenvolvimento do projeto. Ao sistema foi dado o nome de CDD, em referência à comissão que faz a distribuição das disciplinas.

4.2.2 Planejamento

Após o fim da definição do escopo do projeto, deu-se início ao levantamento detalhado dos requisitos do software. Nesta etapa, para se ter um melhor entendimento das necessidades que o projeto deveria atender, foi analisado como o processo de distribuição de disciplinas era realizado até então (Figura 5) e realizado o levantamento dos requisitos funcionais e não funcionais.

Disciplinas	GBC014 - Programação Procedimental	GBC015 - Introdução a Ciência da Computação	GBC016 - Lógica para Computação
Horários	2cd+3cd+6cd	3eq	4cd+5cd
Docentes em 2014/02	MARIA ADRIANA VIDIGAL DE LIMA (4/4) MARCELO ZANCHETTA DO NASCIMENTO (2/4)	IVAN SENDIN (2/4)	JOÃO NUNES DE SOUZA (5/4)
Docentes Atuais 2015/01	MARCELO ZANCHETTA DO NASCIMENTO (3/4) RENAN G. CATTELAN (2/4)	IVAN SENDIN (3/4)	JOÃO NUNES DE SOUZA (6/4)
	MARCELO ZANCHETTA DO NASCIMENTO (4/4) C RENAN G. CATTELAN (3/4) V	MARIA ADRIANA VIDIGAL DE LIMA (1/4)	RITA MARIA DA SILVA JULIA (1/4)

FILAS			
1	ANILTON JOAQUIM DA SILVA	1	ANILTON JOAQUIM DA SILVA
2	AUTRAN MACEDO	2	AUTRAN MACEDO
3	RITA MARIA DA SILVA JULIA	3	RITA MARIA DA SILVA JULIA
4	JAMIL SALEM BARBAR	4	ILMERIO
5	CARLOS LOPES	5	STÉPHANE JULIA
6	ILMERIO	6	MARCELO MAIA
7	CLAUDIO CAMARGO RODRIGUES	7	LUIZ GUSTAVO ALMEIDA MARTINS
8	LUIF FERNANDO FAINA	8	JOSÉ GUSTAVO DE SOUZA PAIVA
9	DENISE GULIATO	9	FLÁVIO DE OLIVEIRA SILVA
10	STÉPHANE JULIA	10	RENAN G. CATTELAN
11	MARCELO MAIA	11	PAULO RODOLFO DA SILVA LEITE COELHO
12	LUIZ GUSTAVO ALMEIDA MARTINS	12	FABIANO DORÇA
13	JOSÉ GUSTAVO DE SOUZA PAIVA	13	BRUNO A.N. TRAVENÇOLO
14	RONALDO CASTRO DE OLIVEIRA	14	ALEXSANDRO SANTOS SOARES
15	RIVALINO MATIAS	15	MAURICIO ESCARPINATI
16	PAULO RODOLFO DA SILVA LEITE COELHO	16	ANDRÉ BACKES
17	FABIANO DORÇA	17	ELAINE R. de FARIA
18	BRUNO A.N. TRAVENÇOLO	18	PAULO HENRIQUE RIBEIRO GABRIEL
19	ALEXSANDRO SANTOS SOARES	19	GINA MAIRA BARBOSA DE OLIVEIRA
20	MAURICIO ESCARPINATI	20	MARCELO ZANCHETTA DO NASCIMENTO
21	ANDRÉ BACKES	21	SHIGUEO NOMURA
22	ELAINE R. de FARIA	22	
23	LÁSARO JONAS CAMARGOS	23	
24	RAFAEL PASQUINI	24	
25	MARIA CAMILA NARDINI BARIONI	25	
26	DANIEL DUARTE ABDALA	26	
27	HUMBERTO LUIZ RAZENTE	27	
28	CHRISTIANE REGINA SOARES BRASIL	28	
29	PRISCILA CRISTINA BERBERT RAMPAZZO	29	
30	PAULO HENRIQUE RIBEIRO GABRIEL	30	
31	IVAN SENDIN	31	
32	GINA MAIRA BARBOSA DE OLIVEIRA	32	
33	RENATO APARECIDO PIMENTEL DA SILVA	33	
34	MARIA ADRIANA VIDIGAL DE LIMA	34	
35	DANIEL FURTADO	35	
36	SHIGUEO NOMURA	36	

Figura 5 – Planilha utilizada para distribuição de disciplinas.

A seguir são listados os requisitos funcionais, que descrevem as funcionalidades desejadas:

- O software deve permitir a entrada no sistema por um professor, utilizando o seu código SIAPE (código utilizado para identificação do funcionário público federal);
- Deverá ser possível visualizar e alterar (inserir e remover) a lista de disciplinas do professor;
- O software deve disponibilizar uma lista de turmas de disciplinas do professor para realizar alterações na ordem de prioridade que o docente deseja ministrar no próximo semestre;
- Deverá ser possível visualizar a fila de professores da disciplina selecionada;
- A fila de professores por disciplina deverá conter as informações do nome do professor, posição na fila, prioridade atribuída para a disciplina, quantidade de vezes que a disciplina foi ministrada pelo professor e o total de vezes que o docente pode ministrar a respectiva disciplina;
- Deverá ser possível visualizar a fila de disciplinas de um professor;
- A fila de disciplinas por professor deverá conter as informações do nome da disciplina, curso, prioridade atribuída a disciplina pelo professor, posição do docente na fila, quantidade de vezes que a disciplina foi ministrada pelo professor e o total de vezes que o docente pode ministrar a respectiva disciplina;
- Deverá ser possível visualizar a fila de professores por uma turma selecionada;
- A fila de professores por turma deverá conter as informações do nome do professor, posição na fila, prioridade atribuída para a turma, quantidade de vezes que a turma foi ministrada pelo professor e o total de vezes que o docente pode ministrar a respectiva turma;
- Deverá ser possível visualizar a fila de turmas por um professor selecionado;
- A fila de turmas por professor deverá conter as informações do nome da turma, letra da turma, curso, prioridade atribuída a turma pelo professor, posição do docente na fila, quantidade de vezes que a turma foi ministrada pelo professor e o total de vezes que o docente pode ministrar a respectiva turma;
- O software deverá possuir um quadro de horários para permitir verificar e comparar os diferentes horários existentes nas turmas pelo professor;
- Deverá ser possível selecionar diferentes semestres no software para a exibição de um histórico de informações;

- O sistema deverá desabilitar as opções de realizar alterações nos dados quando o semestre selecionado for diferente do semestre ativo;

A seguir são exibidos os requisitos não funcionais desejáveis referente ao uso da aplicação:

- O sistema deverá apresentar uma interface simples e intuitiva para o usuário;
- Deverá ser possível acessar o sistema em diferentes navegadores e sistemas operacionais;
- O sistema deverá exibir dados consistentes ao usuário;

A fim de complementar os requisitos do software, foi desenhado um diagrama de casos de uso ([Figura 6](#)) para exemplificar o relacionamento dos professores (atores) com o sistema proposto.

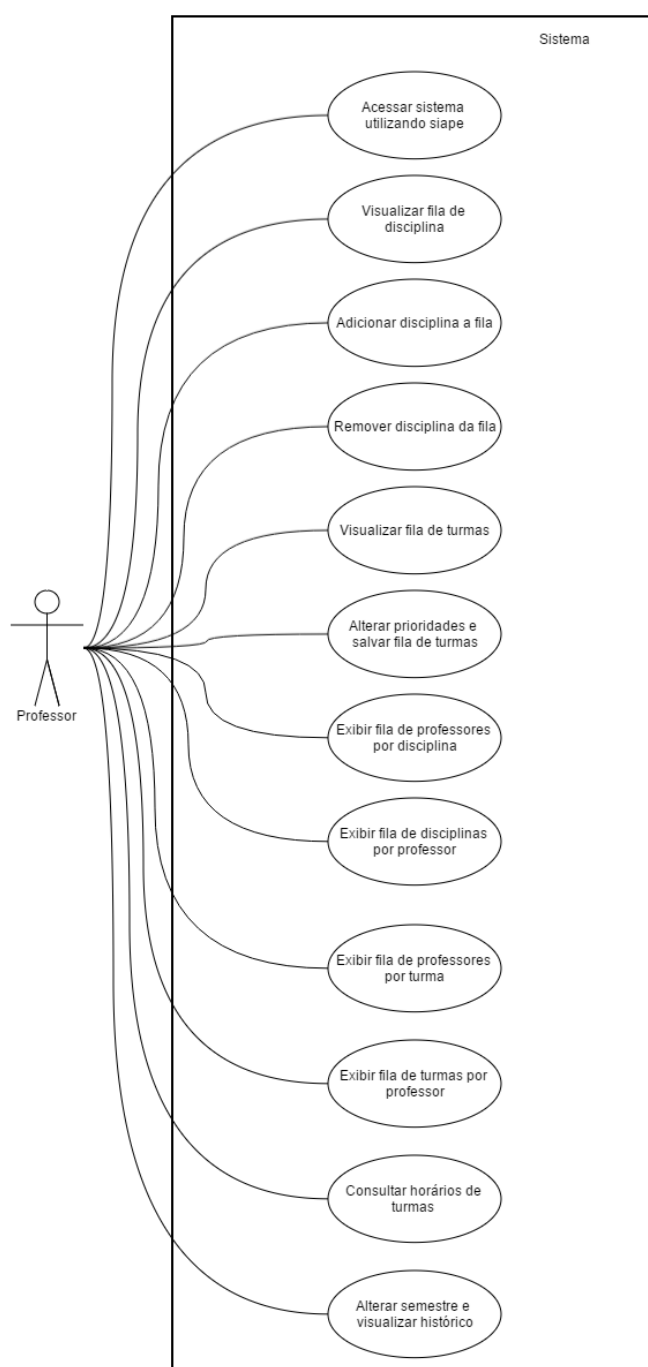


Figura 6 – Diagrama de casos de uso dos requisitos do sistema.

4.2.3 Execução

Na etapa de execução, o software foi desenvolvido utilizando as ferramentas descritas na sessão 4.1, buscando atender todos os requisitos levantados. A construção do projeto aconteceu em três etapas, sendo a construção do modelo de dados, desenvolvimento dos serviços que realizam alterações e consultas na base de dados e desenvolvimento da interface para os usuários.

4.2.3.1 Construção do modelo de dados

Após a análise do processo de distribuição de disciplinas, o modelo de dados (Figura 7) já havia sido desenvolvido, mas foi alterado para atender vários outros requisitos do sistema.

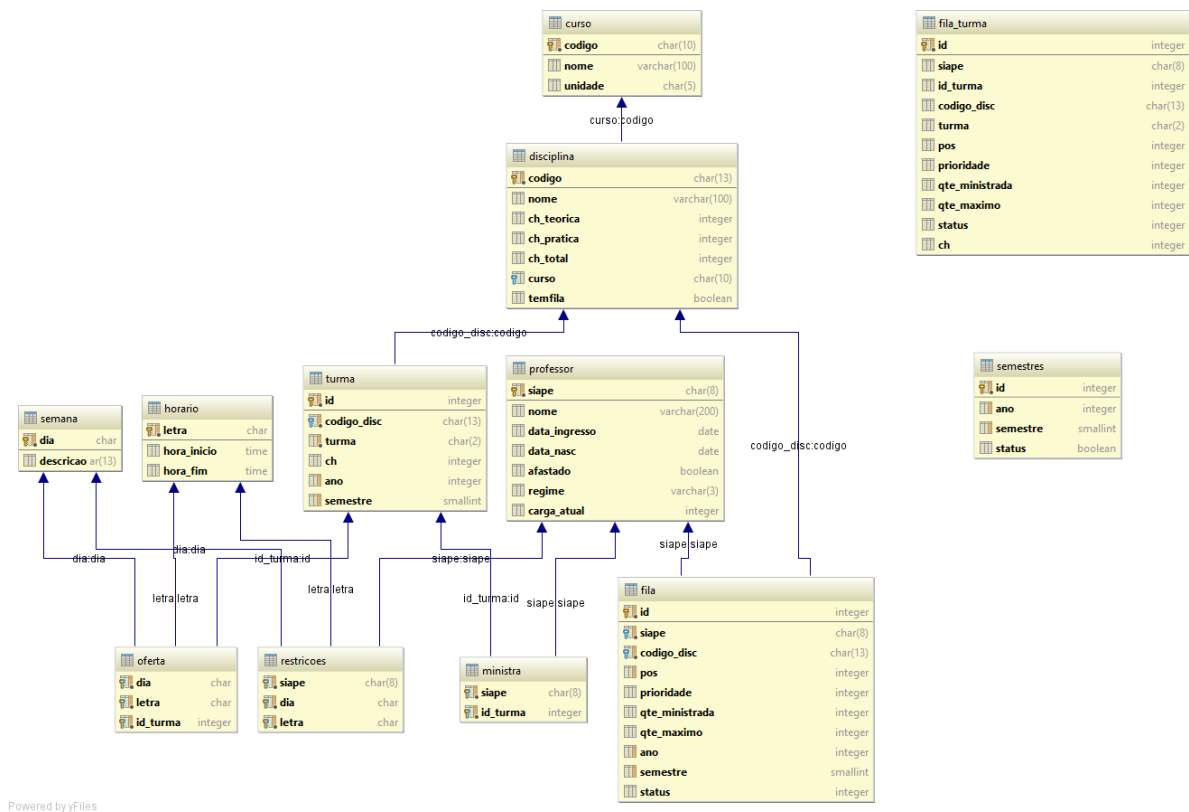


Figura 7 – Diagrama entidade-relacionamento do modelo de dados criado para o sistema.

Com este modelo foi possível inserir todas as informações contidas nas planilhas e manipula-las de maneira mais eficiente, utilizando a linguagem SQL por meio do sistema de banco de dados PostgreSQL.

4.2.3.2 Desenvolvimento dos serviços (“Back end”)

O desenvolvimento do projeto seguiu as orientações do padrão arquitetural MVC, separando as camadas de interface, controle e persistência (Figura 8). O projeto também respeita o paradigma de orientação a objetos visando uma maior organização das funcionalidades existentes.

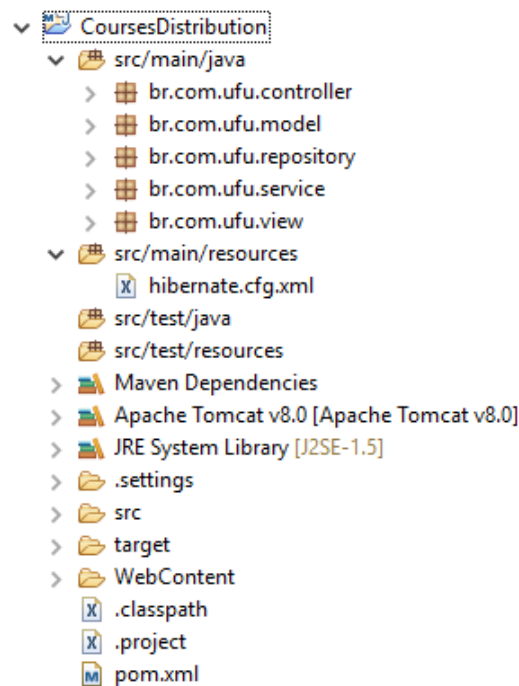


Figura 8 – Organização das classes presentes no software.

O pacote de classes *controller* foi construído com a finalidade de receber requisições, realizar validações e transformações dos parâmetros recebidos em objetos conhecidos do sistema (pacote *view* ou *model*) quando necessário e retornar uma resposta esperada para a camada de visualização. O pacote de classes *service* tem a finalidade de atuar como intermediário entre as requisições recebidas e a camada de persistência. É nesse pacote que se encontram as regras de negócios do sistema. O pacote de classes *model* contém os objetos mapeados do modelo de dados. E o pacote *repository* é a camada de persistência, onde são realizadas todas as operações de pesquisas e alterações no banco de dados do sistema.

Para o funcionamento correto do software existem três arquivos de configurações. O arquivo *pom.xml* é responsável por armazenar as informações de todos os softwares de terceiros sendo utilizados; o arquivo *hibernate.cfg.xml* é responsável por armazenar as configurações de conexão com o banco de dados; e o arquivo *web.xml* tem o papel de realizar a configuração das URLs que são acessadas no software, redirecionando para uma página HTML ou para um serviço nas classes do pacote *controller*.

A [Figura 9](#) exibe o diagrama de classes do sistema e o relacionamento entre os pacotes descritos acima.

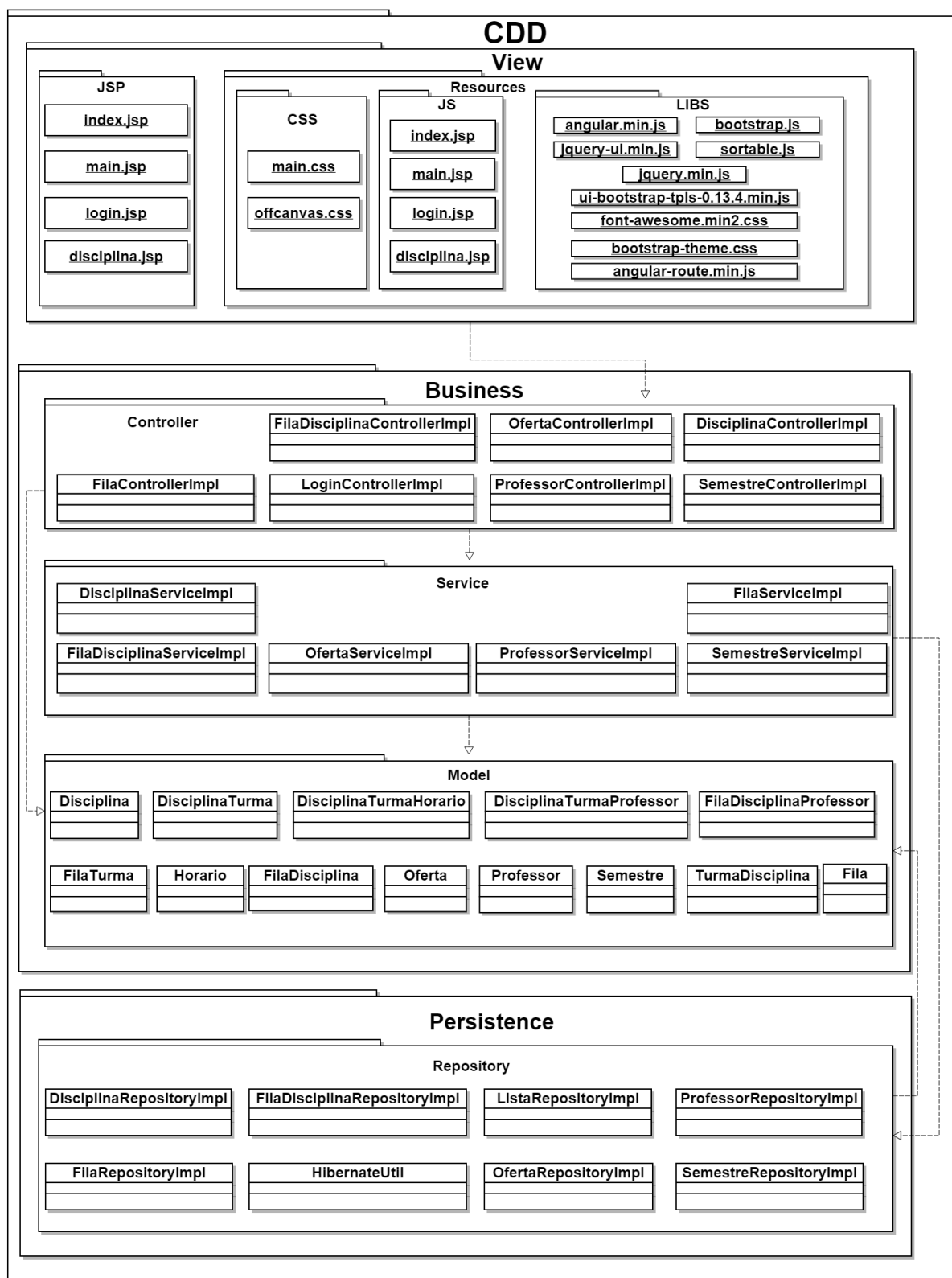


Figura 9 – Diagrama de classes do software.

4.2.3.3 Desenvolvimento da interface web ("Front end")

O desenvolvimento da interface gráfica do sistema aconteceu de forma isolada do

restante do software, com o objetivo de facilitar a criação de novos módulos e alterações futuras. Para a sua construção foi utilizada a linguagem de marcação HTML para a estruturação da página *web*, o *framework* Bootstrap para manipulação da aparência e design da interface e o *framework* AngularJS para deixar o código HTML dinâmico em relação as necessidades de exibição de informações e interação com o usuário.

4.2.4 Encerramento

Após o fim da etapa de execução, o software foi testado em ambiente de produção (servidor localizado na Universidade Federal de Uberlândia) e disponibilizado para uso. Para realizar a instalação do projeto no servidor de aplicações foi utilizado a ferramenta Apache Maven, responsável por gerar o artefato que é executado no servidor *web*.

5 Resultados obtidos

Com o término das etapas de desenvolvimento do projeto, o processo de distribuição de disciplinas ocorreu utilizando a nova ferramenta. A distribuição ocorreu em três etapas: (i) alteração de filas de disciplinas (durante os dias 12 e 18 de novembro); (ii) alteração nas prioridades das disciplinas da fila do docente (durante os dias 24 e 27 de novembro); (iii) visualização de histórico de informações (a partir do dia 28 de novembro).

Durante a etapa de alteração de filas, o usuário era capaz de selecionar quais disciplinas ele gostaria de participar no processo de distribuição para ministrar no semestre 2016-01 (Figura 10). Também era possível visualizar a fila de uma disciplina selecionada ou a fila de um professor selecionado (Figura 11) com informações sobre a prioridade do docente para aquela disciplina, posição do docente na fila, a quantidade ministrada do docente e a quantidade máxima possível.

Figura 10 – Alteração da fila de disciplinas do docente.

Distribuição de Disciplinas
Home

2223307 - Mauricio Amaral Gonçalves
2015 / 2 (Ativo)

Entrar / Sair em filas de disciplinas

Exibir fila por disciplina

Selecione uma disciplina:

FACOM49060 - Sistemas Operacionais - MECATRONIC

EXIBIR LISTA DE PROFESSORES

FACOM49060 - Sistemas Operacionais - MECATRONIC

Professor	Posição na fila	Prioridade	Qte. Min/Qte. Max
Rivalino Matias Jr.	1	1	1/4
Jamil Salem Barbar	2	40	0/4
Luís Fernando Faina	3	7	0/4
Marcelo de Almeida Maia	4	45	0/4
Renan Gonçalves Cattelan	5	12	0/4
Daniel Duarte Abdala	6	9	0/4
Marcelo Zanchetta do Nascimento	7	4	0/4
Paulo Henrique Ribeiro Gabriel	8	37	0/4
Shiguelo Nomura	9	3	0/4

Exibir fila por professor (sem as turmas)

Selecione um professor:

Sílvia Bacalá Júnior

EXIBIR LISTA DO PROFESSOR

Disciplina	Curso	Prioridade	Posição	Qte. Min/Qte. Max
GGI030 - Programação Orientada a Objetos	Gestão da Informação	1	15	0/4
GBC064 - Engenharia de Software	Ciência da Computação	2	2	0/4
GGI026 - Algoritmos e Programação	Gestão da Informação	3	4	0/4
GBC084 - Programação para Internet	Ciência da Computação	4	1	2/4
GGI012 - Oficina de Programação e Laboratório	Gestão da Informação	5	5	0/4
GBC035 - Programação Orientada a Objetos 1	Ciência da Computação	6	4	0/4
GGI024 - Projeto e Desenvolvimento de Software	Gestão da Informação	7	2	0/4

Figura 11 – Consultas para avaliar filas de disciplinas e professores.

Na etapa de alterações de prioridades da fila do usuário, a entrada e saída de filas de disciplinas não era mais possível de ser executada. Para cada disciplina é criada uma ou mais turmas e seus horários são definidos. Nesse ponto, fica liberado para o usuário a alteração de prioridades na sua fila de turmas selecionadas no processo anterior. A [Figura 12](#) exhibe as turmas disponíveis para realizar a alteração de prioridade e a [Figura 13](#) mostra como esse processo pode ser executado arrastando as turmas da fila. Também é possível consultar os horários das turmas para auxiliar no processo de definição de prioridades para as turmas que se deseja ministrar no próximo semestre ([Figura 14](#)) e visualizar todas as informações de filas de disciplinas, turmas e professores de forma semelhante ao descrito na primeira etapa.

Distribuição de Disciplinas
Home

2223307 - Mauricio Amaral Gonçalves
2015 / 2 (Ativo)

Escolher prioridade de disciplinas

SALVAR

1 - GGI038 - Banco de Dados (I) - GESTAO
2 - GGI030 - Programação Orientada a Objetos (I) - GESTAO
3 - GSI015 - Programação Orientada a Objetos 1 (S) - BSI
4 - GSI020 - Programação Orientada a Objetos 2 (S) - BSI
5 - GBC035 - Programação Orientada a Objetos 1 (C) - BCC
6 - GGI041 - Redes de Computadores (I) - GESTAO
7 - GSI023 - Redes de Computadores (S) - BSI
8 - FACOM39801 - Sistemas de Bancos de Dados (I) - GESTAO
9 - GSI016 - Bancos de Dados 1 (S) - BSI
10 - GGI036 - Sistema Operacional (I) - GESTAO
11 - GBC055 - Programação Orientada a Objetos 2 (C) - BCC
12 - GBC043 - Sistemas de Banco de Dados (C) - BCC
13 - GBC056 - Arquitetura de Redes de Computadores (C) - BCC
14 - FACOM39802 - Programação para Internet (I) - GESTAO
15 - GBC084 - Programação para Internet (C) - BCC
16 - GSI019 - Programação para Internet (S) - BSI

Verificar horários das disciplinas

Figura 12 – Layout utilizado para modificar a prioridade das turmas na lista.

Escolher prioridade de disciplinas

SALVAR

2 - GSI015 - Programação Orientada a Objetos 1 (S) - BSI
3 - GGI038 - Banco de Dados (I) - GESTAO
4 - GSI020 - Programação Orientada a Objetos 2 (S) - BSI
1 - GGI030 - Programação Orientada a Objetos (I) - GESTAO
5 - GBC035 - Programação Orientada a Objetos 1 (C) - BCC
6 - GGI041 - Redes de Computadores (I) - GESTAO

Figura 13 – Alteração de prioridade utilizando a funcionalidade “arrastar e soltar”.

Distribuição de Disciplinas

Home

Verificar horários das disciplinas

☐ GG038 - Banco de Dados (I) - GESTAO

☒ GS020 - Programação Orientada a Objetos 2 (S) - BSI

☐ GS023 - Redes de Computadores (S) - BSI

☐ GG036 - Sistema Operacional (I) - GESTAO

☐ GBC056 - Arquitetura de Redes de Computadores (C) - BCC

☐ GS019 - Programação para Internet (S) - BSI

☐ GG030 - Programação Orientada a Objetos (I) - GESTAO

☐ GBC035 - Programação Orientada a Objetos 1 (C) - BCC

☒ FACOM39801 - Sistemas de Bancos de Dados (I) - GESTAO

☐ GBC055 - Programação Orientada a Objetos 2 (C) - BCC

☐ FACOM39802 - Programação para Internet (I) - GESTAO

☐ GS015 - Programação Orientada a Objetos 1 (S) - BSI

☐ GG041 - Redes de Computadores (I) - GESTAO

☒ GS016 - Bancos de Dados 1 (S) - BSI

☐ GBC043 - Sistemas de Banco de Dados (C) - BCC

☐ GBC084 - Programação para Internet (C) - BCC

	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira	Sabado
07:10						
08:00						
08:50						
09:50						
10:40						
11:30						
13:10						
14:00						
14:50						
16:00						
16:50						
17:40						
18:10						
19:00	GS016 -			GS020 - FACOM39801 -	GS020 -	
19:50	GS016 -			GS020 - FACOM39801 -	GS020 -	
20:50		GS016 -		FACOM39801 -		
21:40		GS016 -		FACOM39801 -		

Figura 14 – Funcionalidade para visualizar e comparar horários entre turmas (as turmas que não possuem conflitos em seus horários aparecem em azul e as que possuem aparecem em vermelho).

Na última fase o usuário não pode realizar alterações na base de dados do sistema e fica limitado a apenas visualizar as informações cadastradas durante as duas primeiras etapas e o histórico de informações dos semestres anteriores. A Figura 15 exibe o sistema sem as opções para o professor realizar alterações na sua fila de disciplinas ou nas prioridades das turmas.

Distribuição de Disciplinas	Home
1770125 - Bruno Augusto Nassif Travençolo	
2016 / 1 (Ativo)	
Verificar horários das disciplinas	
Exibir fila por disciplina	
Exibir fila por professor (sem as turmas)	
Exibir fila por turma	
Exibir fila por professor (com as turmas)	

Figura 15 – Sistema apenas com os menus para realizar consultas de informações.

6 Considerações Finais

Podemos evidenciar que a automação de tarefas por meio de sistemas informatizados é uma prática extremamente benéfica, pois visa por meio da utilização da máquina, auxiliar o trabalho humano mecânico. No decorrer do desenvolvimento deste projeto, as maiores dificuldades encontradas foram duas alterações nos requisitos que tiveram grande impacto nas regras de negócio do sistema.

Mesmo após o termino da implantação do software, existe a necessidade de criação de novos módulos, como o modulo administrativo para facilitar na configuração e grandes alterações de dados no sistema. Outro modulo necessário é o modulo de geração de relatórios para retirada de informações estatísticas sobre as informações existentes.

O objetivo deste trabalho foi facilitar o processo de distribuição de disciplinas estruturando os dados existentes em um modelo de banco de dados relacional e prover uma interface amigável aos usuários para a manipulação dessas informações. Vários professores que utilizaram o sistema relataram que o uso do mesmo foi muito fácil e intuitivo. Durante seu desenvolvimento, todos os requisitos previamente levantados foram cumpridos e a data limite de termino do projeto foi respeitada.

Referências

ANGULARJS. *Developer Guide*. 2015. <<https://docs.angularjs.org/guide/introduction>>. [Online; accessed 13-September-2015]. Citado na página 18.

BOOTSTRAP. *Get Bootstrap*. 2015. <<http://getbootstrap.com/>>. [Online; accessed 13-September-2015]. Citado na página 18.

INSTITUTE, P. M. *Um guia do conhecimento em gerenciamento de projetos*. [S.l.]: Editora Saraiva, 2012. Citado 6 vezes nas páginas 4, 12, 13, 14, 15 e 16.

NETBEANS. *Introdução aos Web Services RESTful*. 2015. <https://netbeans.org/kb/docs/websvc/rest_pt_BR.html#create-services-from-db>. [Online; accessed 01-November-2015]. Citado na página 18.

W3SCHOOLS. *The MVC Programming Model*. 2015. <http://www.w3schools.com/aspnet/mvc_intro.asp>. [Online; accessed 17-July-2015]. Citado 2 vezes nas páginas 4 e 11.