

**IBTA-BILAC** - INSTITUTO BRASILEIRO DE TECNOLOGIA AVANÇADA   
DE SÃO JOSÉ DOS CAMPOS

Ralph **ANGELI**

Flávio **NOGUEIRA**

João Pedro Silvério **SILVA**

COURSE FACTORY SYS - Sistema de Oferta de Disciplinas

Ralph **ANGELI**

Flávio **NOGUEIRA**

João Pedro Silvério **SILVA**

COURSE FACTORY SYS - SISTEMA DE OFERTA DE DISCIPLINA

Trabalho de Conclusão de Curso, apresentado à Faculdade Bilac para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Gerson Penha Neto

**AQUI VAI UMA PÁGINA COM REFERÊNCIAS QUE PRECISAM SER VERIFICADAS COM O GERSON O QUE SÃO.**

Ralph **ANGELI**

Flávio **NOGUEIRA**

João Pedro Silvério **SILVA**

COURSE FACTORY SYS - SISTEMA DE OFERTA DE DISCIPLINA

Trabalho de Conclusão de Curso, apresentado ao

IBTA-Bilac Instituto Brasileiro de Tecnologia

Avançada para a obtenção do título de Bacharel.

Aprovado em \_\_\_\_ / \_\_\_\_ / \_\_\_\_

BANCA EXAMINADORA

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Me. Gerson Penha

IBTA-Bilac Instituto Brasileiro de Tecnologia Avançada

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Dr. Marcos Cunha

IBTA-Bilac Instituto Brasileiro de Tecnologia Avançada

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Me. Marcos Reis

IBTA-Bilac Instituto Brasileiro de Tecnologia Avançada

**Dedicado a todos àqueles que acreditam em nós.**

**AGRADECIMENTOS**

Agradecemos nossas famílias e amigos que nos incentivaram durante todo o desenvolvimento deste projeto e que nos deram energia para seguir em frente e superar obstáculos. Agradecemos também ao professor Gerson, nosso orientador e suporte durante a realização deste trabalho, e a todos os demais professores que nos apoiaram durante este semestre.

O que a mente é capaz de conceber e acreditar,

ela é capaz de criar!

**Napoelon Hill**

**RESUMO**

Após a entrevista com o diretor de uma instituição de ensino foi relacionado que a maneira como eram escolhidas as disciplinas a serem ofertadas nos cursos superiores, era um processo robusto de grande demanda de carga horária dos envolvidos no processo e que a análise dos dados tornavam-se imprecisas, sendo o fator principal desse fato o método utilizado pela instituição no período da entrevista: planilhas distintas.

Devido a este ser um ponto crítico para gestão dos processos da escolha das disciplinas a serem disponibilizadas em cada curso da instituição relacionada acima, foi-se desenvolvido o sistema de oferta de disciplinas nomeado por Course Factory SYS, de modo a tornar este processo em algo que possa ser realizado de maneira eficaz e eficiente, com melhor qualidade e precisão na geração resultados concisos.

Neste projeto é descrito sobre o ambiente que dispõe da facilidade de visualizar, escolher e adaptar a grade de disciplinas a serem ofertadas aos alunos de uma faculdade, facilitando assim o processo de distribuição curricular do aluno, garantindo gerenciamento qualificado e preciso.

O sistema é composto pelas telas de inserção, manipulação e exclusão de dados, de modo a alimentar o sistema para que em uma segunda etapa do sistema haja a compilação desses dados e seja resultante ao usuário as informações das melhores disciplinas a serem ofertadas naquele período (semestre ou trimestre).

**Palavras-chaves:** Disciplina, Faculdade, sistema de oferta de disciplinas, melhores disciplinas a serem ofertadas.

***Abstract***

*Lorem ipsum dolor sit amet, posuere quam cum varius eu mi, tristique mollis ipsum varius purus et. Sagittis neque nulla. Nibh integer vestibulum aenean odio. Mauris pharetra, eaque mauris vel magna varius, magnis mauris mus. Penatibus praesent integer venenatis sed in, felis vel orci per, diam erat.*

*Semper scelerisque fusce quisque. Erat adipisicing rhoncus morbi in, malesuada pede eu ut ornare, ultricies nibh penatibus convallis lectus lacinia, erat gravida. Nonummy pede lorem in, tempus egestas cum, vel libero. Turpis in wisi pede, donec wisi suspendisse orci erat iaculis, nec lorem id nulla, lorem tincidunt condimentum ac maecenas. Donec rutrum id mollis dui. Aliquam ut maecenas expedita morbi pellentesque, morbi nunc purus suscipit lobortis donec mattis, et ac eleifend in sit ut diam. Pede consectetuer dui sit. Donec sit a laoreet non consequat, magna id felis, tempor ultricies, diam in. Gravida pulvinar vitae aliquam, turpis laoreet nullam, neque mattis. Sed laoreet, enim malesuada. Consectetuer wisi integer, arcu placerat neque quam proin incidunt.*

***Keywords****: Lorem, ipsum.*

# 

**Índice de Figuras**

*Figura 1: Fases do processo de compilação.*

*Figura 2: Elementos do sistema de Pattis.*

# 

# 

**ÍNDICE DE TABELAS**

*Tabela 1: Tradução de Código-Objeto padrões do compilador.*

*Tabela 2: Tradução de Código-Fonte para Código-Objeto.*

**LISTA DE ABREVIATURAS E SIGLAS**

**BRASSCOM** – Associação Brasileira das Empresas de Tecnologia da Informação e Comunicação

**CSS** – *Cascading Style Sheets*

**EBNF** *– Extended Backus-Norm Form*

**GPL** – *General Public License*

**HTML** – *Hyper Text Markup Language*

**IDC** – *International Data Corporation*

**MCTI** – Ministério de Ciência, Tecnologia e Inovação

**TI** – Tecnologia da Informação

**WHATWG** – *Web Hyper Text Application Technology Working Group*

**W3C** – *World* *Wide* *Web* *Consortium*

# 

**Sumário**

**1** **Introdução**

**2** **Fundamentação Teórica**

2.1 Gramática e Linguagem

2.2 Compiladores

2.2.1 Análise Léxica

2.2.2 Análise Sintática

2.2.3 Análise Semântica

2.2.4 Código Intermediário e Código-Objeto

**3** **Karel**

3.1 O Robô, suas Capacidades e Limitações.

3.2 Programa Exemplo

**4** **Projeto**

4.1 Ambiente

4.2 Gramática

4.3 Estrutura do Compilador

4.4 Diagramas Sintáticos

4.4.1 Programa

4.4.2 Definição

4.4.3 Identificador

4.4.4 Expressão

4.4.5 Bloco

4.4.6 Repita

4.4.7 Enquanto

4.4.8 Condicional

4.4.9 Instrução

4.4.10 Testes

4.5 Exemplos da Análise Léxica

4.6 Código-Objeto

4.7 Exemplo de Código-Fonte e respectivo Código-Objeto

4.8 Manual do Usuário

4.8.1 Ambiente de Programação

4.8.2 Editor de Mapas

4.9 Disponibilização do Ambiente

**5** **Tecnologias**

5.1 Bootstrap

5.2 CSS3

5.3 Git

5.4 GitHub

5.5 Google Chrome

5.6 HTML5

5.7 Inkscape

5.8 Internet Explorer

5.9 JavaScript

5.10 jQuery

5.11 Mozilla Firefox

5.12 Node.JS

5.13 Pixi.JS

5.14 Safari

5.15 Sublime Text

**6** **Conclusão**

**REFERÊNCIAS**

# Introdução

Segundo a Faculdade Univel, um “curso de graduação” é a denominação dada aos cursos de nível universitário, em que esse curso se enquadra sempre em algumas das três seguintes categorias: Licenciatura, Bacharelado ou Tecnólogo.

Dessa maneira, tem-se que Licenciatura apresenta matérias de cunho pedagógico nos cursos, preparando o estudante principalmente para atuação na área docente do ensino fundamental e médio. O bacharelado, em suma, abrange uma grande gama das vertentes de determinada área escolhida, aferindo um conhecimento sobre diversos assuntos. Diferentemente, o curso de tecnologia se forma com o foco na atuação de uma área específica, apresentando normalmente menor tempo de duração em relação aos demais. (InfoEscola 2015).

Dentro dos aspectos acima, existem ainda os cursos presenciais, em que o aluno deve ter o mínimo de 75% de presença nas atividades escolares (aulas, provas, etc). Existe o curso EAD, em que não é obrigatória a presença do aluno e todas ministrações de aula, assim como os materiais são oferecidos diretamente pela internet, o que também dá origem para uma terceira ramificação, que é o curso Semi-Presencial, em que as aulas são ministradas tanto online, como também presencialmente, em alguns dias da semana. (Portal Brasil, 2009).

Tendo em vista quais são as possíveis classificações a que um curso superior se adequa, tem-se que estes são divididos em módulos, normalmente cursados em períodos trimestrais ou semestrais, e em cada módulo existe uma grade de disciplinas que devem ser cursadas.

Algumas disciplinas possuem pré-requisitos. Segundo a Universidade Católica de Petrópolis (UCP), um pré-requisito, trata-se de uma disciplina que tenha sido cursada com aprovação, para que assim haja o embasamento necessário para a matricular-se em outras disciplinas de períodos seguintes.

Assim, para que haja o gerenciamento das disciplinas a serem ofertadas em determinado período letivo, de maneira organizada e concisa, é devido que a instituição utilize de um sistema que atenda às suas regras de negócio, junto aos critérios essenciais caracterizados acima. Pois, gerir as disciplinas que serão ofertadas em um período letivo, é algo peculiar para uma instituição de ensino superior, uma vez que essa tarefa passa a não ser algo trivial.

Isso ocorre por diversos motivos, seja pelo número de alunos, que poder ser grande em uma época e pequena em outra; pelos trancamentos de matrículas; reprovações dos alunos, assim ainda como o reaproveitamento de disciplinas iguais para cursos e series diferentes (também conhecido como enlaçamento); disponibilidade de professores; entre outros fatores que tornam essa atividade em algo dependente de variantes.

Esse gerenciamento é algo que deve ser feito de maneira minuciosa, de modo a não afetar a grade curricular do aluno,mas que garantam a melhor forma de aproveitamento do aluno e a viabilidade financeira da instituição.

No momento em que foi realizado o levantamento junto ao cliente, sobre qual a ferramenta utilizada, foi relatado que este gerenciamento era feito manualmente através de planilhas, e, trimestralmente, os coordenadores de cursos e os auxiliares de coordenação, avaliavam a planilha para analisar as disciplinas que seriam ofertadas no período seguinte, levando como base os diversos parâmetros para realização de tal atividade.

Além do processo ser feito manualmente, era demandando o investimento de grande quantidade de horas por parte dos colaboradores na realização dessa atividade, e ainda assim com a possibilidade de não alcançar os melhores resultados ao final do processo.

Diante do cenário exposto, o presente trabalho apresenta o sistema web Course Factory SYS, que permite a instituição de ensino visualizar o histórico de ofertas de disciplinas, a gerir as características dos cursos e sugerir as próximas ofertas, baseadas em parâmetros definidos pela faculdade.

# Ciclo de Vida

Para que se haja o maior controle sobre os processos do projeto, foi definido seu ciclo de vida, de modo a compreender e ter melhor visibilidade sobre cada etapa do sistema. Nesse método, o projeto foi divido em fases, sendo elas: Concepção, Desenvolvimento, Operação, Manutenção e Descarte. Assim, cada uma das etapas podem ser descritas da seguinte maneira.

Concepção é onde compreende, com maior rigor, o ponto a se realizar a análise dos requisitos do sistema (definição do escopo), assim como a perspectiva do ciclo de vida do projeto. Essa fase ainda compreende a etapa de preparação para o desenvolvimento do projeto (análise técnica), assim como definição de layout, e proposta para aprovação junto ao cliente, que é seguido pela fase de “Desenvolvimento”.

Na fase de Desenvolvimento, são determinados os cenários de produção do sistema, como o planejamento de desenvolvimento (metodologia, metas, etc), também o desenvolvimento do front-end como do back-end, sendo concluída a fase de desenvolvimento no ponto em que é realizada a implementação do sistema, já garantindo que o sistema esteja em perfeito funcionamento.

Uma vez que o sistema esteja em perfeito funcionamento, está apto para a fase de “Operação”, ponto este que é caracterizado pode ser onde são executadas as tarefas exigidas pelo cliente, nesse caso, permitido fazer toda manipulação de dados e operações do sistema, como determinadas na etapa de Concepção.

A fase de “Manutenção” que cabe detectar e corrigir falhas, e a fase de “Descarte” são cabíveis quando, respectivamente, o sistema apresenta alguma situação divergente ao que era pretendido ou torna-se obsoleto.

# Concepção de entidades

Para que se possa ter uma compreensão mais clara sobre alguns fatores internos de maior importância do sistema, a Tabela 1.0 propõe expor a concepção de cada entidade.

|  |  |  |
| --- | --- | --- |
| **Entidade** | **Atributos** | **Descrição** |
| Curso | * ID * Nome * Abreviação * Quantidade Módulos * Data de aprovação MEC * ID-MEC * Duração | Representação de um Curso real e suas características.  Exemplo:  Ciência da Computação, é um curso, de onde ramificarão turmas. |
| Turma | * ID * Nome * Período Letivo * ID\_Curso | Representação de uma Turma real e suas características.  Exemplo:  “Décima Quinta Turma de Bacharelado em Ciênca da Computação” , toda turma tem um nome de identificação único |
| Disciplina | * ID * Nome * Abreviatura * Carga Horária * Pré Requisitos * Descrição | Representação de uma Disciplina a ser ministrada e suas características.  Exemplo:  “Paradigmas de Linguagens de Programação” e essa que por sua tem como Pré Requisito a disciplina “Algoritmos” |
| Aluno | * ID * Nome * Sobrenome * RA * ID\_Turma * CPF * Bolsa * Financiamento * Data Matrícula | Registro do aluno, o qual é relacionado a uma turma que tem vínculo com curso e às disciplinas atreladas a este curso. |

Tabela 1.0 - Concepção de Entidade

# Análise dos requisitos

A análise dos requisitos é a etapa onde é realizada a observação e levantamento do ambiente no qual será implantado o sistema. Para isso deve-se levar em consideração os fatores como: usuário do sistema, a motivação da operação e os resultados finais. (Linha de Código).

Para a análise dos requisitos do sistema, é preciso compreender a habilidade da Modelagem de Negócio, onde, segundo Rodrigues, 2010 em publicação para o portal Devmedia, a modelagem de negócio relaciona-se com o aspecto de tornar conhecido o negócio do cliente, seu ramo direto de atuação, sua necessidade, seus maiores problemas e de que forma eles podem se relacionar ao sistema que será desenvolvido, direta ou indiretamente. Esse é um processo de suma importância para que haja concordância entre o sistema desenvolvido e os objetivos deste negócio.

Nesse contexto, para a compreensão dos elementos necessários para o desenvolvimento do sistema foi analisado junto ao Cliente as métricas de interesses de cada Stakeholder. Segundo Felipe Godoy em Tracto - 2013, Stakeholder é a pessoa, ou grupo de pessoas que interagem e afetam direta ou indiretamente o ciclo de vida do sistema.

## 

## 3.1. Requisitos Funcionais

Segundo Sommerville, 2008, os requisitos funcionais são as especificações de como o sistema deve se comportar em determinadas situações, em suma os requisitos funcionais estabelecem a visão do que o sistema deve fazer. Estes requisitos dependem dos Stakeholders e de qual tipo de software está sendo desenvolvido.

Desse modo, o fluxo de sequência (simples) se dá na seguinte forma:

1. Entrar do sistema
2. Selecionar “Cadastro de Nova Disciplina”
   1. Preencher informações da disciplina
   2. Pressionar “Salvar”
3. Selecionar “Cadastro de Novo Curso”
   1. Preencher informações do curso
   2. Pressionar “Salvar”
4. Selecionar “Cadastro de Nova Turma”
   1. Preencher informações da turma
   2. Pressionar “Salvar”
5. Selecionar “Cadastro de Novo Aluno”
   1. Preencher informações do aluno
   2. Pressionar “Salvar”
6. Selecionar “Gerar Quadro”
7. Pressionar botão “Gerar”
8. Emitir relatório com os resultados
9. Finalizar sistema.

Observações: As etapas 2, 3, 4, e 5 podem ser repetidas quantas vezes forem necessárias.

Fluxo de exceção: Caso haja alguma restrição nas etapas 3, 4, ou 5, deve se verificar se as etapas anteriores forem seguidas corretamente, para isso deve-se utilizar das informações prestadas em “Consultar Disciplina”, “Consultar Curso”, “Consultar Turma” ou “Consultar Aluno”. Para escolher qual consulta deve ser realizada, é preciso que se atente em que ponto o erro ocorreu e qual o passo anterior.

Exemplo: Quando o erro é dado a partir do ponto 3 (Cadastro de Novo Curso), deve-se

1. Selecionar “Cadastro de Novo Curso”
   1. Preencher informações do curso
   2. Pressionar “Salvar”
   3. Mensagem de erro
      1. Identificar se erro é dado por equívoco no preenchimento, ou por pré-requisito. (Se erro é dado por preenchimento equivocado, deve-se repetir o ponto 3.a. Caso o erro tenha se dado por conta de algum pré-requisito, deve-se seguir o fluxo).
   4. Pressionar “Voltar”
2. Selecionar “Consultar Disciplina”
   1. Verificar se informações foram preenchidas corretamente
      1. Alterar o que for necessário
   2. Pressionar botão “Salvar”
3. Repetir passo 3.

Como requisitos funcionais do sistema, são destacados no CFS os seguintes requisitos:

* CRUD - do inglês, o *Create, Read, Update and Delete* (criar, ler, alterar e deletar) - de cursos, turmas, disciplinas e alunos.
* Todas entidades estão relacionadas às disciplinas, onde:
  + Curso é composto por turm
* Cada aluno deve ter relacionado quais matérias foram cursadas, e em quais destas ele está aprovado ou reprovado.
* O cada aluno deve ter a marcação caso esteja “ativo” naquele período letivo, ou “Inativo” se estiver com a matrícula trancada. Para que dessa maneira possa se fazer o levantamento de quantos alunos podem cursar determinada matéria naquele período.

O principal requisito fundamental do sistema compreende verificar quais as melhores disciplinas a serem ofertadas no corrente período letivo, seja trimestral ou semestral, para tal, o sistema deverá verificar algumas variantes, decorrentes das regras de negócio da Faculdade Bilac, denotada a partir desse ponto por Cliente:

1. Quantidade de alunos que poderão cursar aquela determinada disciplina;
   1. Uma mesma disciplina pode ser ofertada a diversos cursos, portanto é preciso verificar se aquela disciplina apresenta algum pré-requisito para algum curso. Caso a disciplina tenha pré-requisitos, é preciso verificar se o aluno já está apto a cursá-la, ou seja, se ele já cumpriu com êxito a disciplina base para a outra.
   2. As disciplinas que puderem comportar o maior número de alunos será a mais indicada para ser ofertada.
2. Tempo sem ofertar disciplina
   1. Quando determinada disciplina fica mais de 4 trimestres sem ser ofertada, ela deverá ter prioridade em ser ofertada, mesmo que não preencha os requisitos do ponto 1.2 e tenha baixo número de alunos que poderão cumprir com aquela matéria.

**Requisitos Não Funcionais**

Sommerville define como requisitos não funcionais aqueles requisitos que não estão diretamente ligados com as funções especificas, elas estão ligas as situações emergentes do sistema e podem definir restrições, especificações da aplicação, surgiram devido a necessidade do usuário, como restrições de orçamento, politicas organizacionais e operabilidade com outros softwares e hardwares. Os requisitos não funcionais estão diretamente relacionados a qualidade do sistema. Exemplos de requisitos não funcionais são os requisitos de Usabilidade, Confiabilidade, Implementação, Segurança e Permissão.

**Engenharia de Software**

Segundo Fritz Bauer, a Engenharia de software é pautada na criação e utilização de sólidos princípios de engenharia com o objetivo de se obter um software de maneira econômica, onde isso se relaciona com baixo custo operacional para desenvolvimento, e agilidade para que esse desenvolvimento será realizado, (posteriormente adotado as metodologias ágeis de desenvolvimento), e que ainda haja uma compreensão sobre a qualidade do sistema em questão, abordando diretamente a eficiência, confiabilidade e disponibilidade do sistema, e que esse atenda a operacionalidade em máquinas reais.

Sommerville, 2008 define a engenharia de software como o princípio da engenharia em todos aspectos da produção de um software, desde a fase de concepção do projeto até o ponto de sua manutenção. Ainda, Pressman, 2006, complementa em seu livro dizendo que sobre tudo, na Engenharia de Software a “pedra fundamental” é a cultura do aperfeiçoamento pautado na qualidade do sistema.

**Scrum**

Segundo Schwaber, 2009, Scrum é fundamentado na teoria de controle de processos baseados nos resultados gerados a partir da abordagem iterativa e incremental, visando assim otimizar a previsibilidade de possíveis riscos no seu desenvolvimento, podendo os tratar com maior efetividade. Para isso o Scrum é desenvolvido com base em três pilares, sendo eles: Transparência (I), Inspeção (II) e Adaptação (III).

|  |  |
| --- | --- |
| **Pilar** | **Característica** |
| **Transparência** | Garantia de que os resultados estejam visíveis ao gerente de projeto, de modo em que este possa compreender os pontos alcançados e os próximos passos a serem dados |
| **Inspeção** | Para que possíveis falhas e incoerências do projeto sejam tratadas eficazmente, a rotina de inspeção é levada em consideração de maneira muito criteriosa. |
| **Adaptação** | Com base na inspeção, o gerente do projeto deve determinar quais pontos que deverão ser readequados, assim, fazendo com que não haja desperdício de tempo com atividades menos importantes para cada momento do projeto. |

Tabela 2.0 - Pilares do SCRUM

No SCRUM é formado com a elaboração de grupos, aqui chamados de Time Scrum, isso para que possa se otimizar a produtividade e flexibilidade do projeto. De acordo com Schwaber, 2009, o time scrum é formado por três funções principais funções: 1) ScrumMaster, responsável garantir que o processo de desenvolvimento seja compreendido e seguido; 2) Product Owner, que é responsável por agregar e maximizar valor ao trabalho feito pelo Time; 3) O time, propriamente dito, no qual encontra-se os responsáveis em desenvolver o que é proposto, a fim de ter um pedaço potencialmente entregável do produto, ao final do Sprint.

Segundo Batista, em publicação pelo portal Devmedia, para que se tenha a referência do que se deve ser feito, é criado uma espécie de lista chamada de “Product Backlog”, onde deverá conter todas necessidades definidas pelo Product Owner, e assim compreender os processos a serem desenvolvidos. Seguido deste, deve-se ter o “Sprint Backlog”, que deve ser formulado como se fosse uma “sub-lista”, para que se possa formular o “Sprint”. Almeida Dias, 2013, declara que um Sprint é iniciado pelo planejamento detalhado das tarefas a serem executadas em determinado período de tempo, normalmente de 7 a 30 dias. Schwaber declara que nesse processo o ScrumMaster deve garantir que nenhuma mudança que possa afetar a meta da Sprint será feita, assim assegurando o cumprimento do prazo de entrega e qualidade. No inicío das Sprints, é feita uma reunião de Planejamento de Sprint, e para que seja legitimado o trabalho de desenvolvimento, são realizadas as denominadas “Daily Scrum” em que são verificadas atividades desenvolvidas no dia anterior, o que está desenvolvendo no dia decorrente e se há algo que o possa estar atrapalhando na atividade em questão. Ao final do tempo estimado da Sprint, antes que seja realizado o planejamento de uma nova Sprint, são feitas a Revisão da Sprint, que detalhará o que feito, e o que não foi concluído e seu respectivo motivo, e a Retrospectiva da Sprint que avaliará os fatores positivos e negativos da Sprint anterior. Desse modo, ao final de cada Sprint, é comum que se haja uma apresentação ao cliente de uma versão prévia do sistema.

**Roadmap**

Baseado nas palavras de Toledo, 2012, um Roadmap, dentro de um projeto como o CFS, é criado a partir da definição dos requisitos do sistema, identificando assim quais os principais objetivos e metas para o sistema a curto, médio e longo prazo.

O Roadmap deve conter de maneira sucinta os principais passos que serão dados, seja expansão do projeto, desenvolvimento de versão melhorada entre outros, normalmente para isso, conta-se com as metodologias de desenvolvimento ágil, pois então se haverão métricas para seguir.

Tratando-se de Roadmap é comum que existam alguns pontos que deverão ser modificados no decorrer do tempo. Com isso, no Roadmap existe um estudo chamado de “Cone da Incerteza”, que trata as incertezas da relação de tempo e resultado. [Guia da Startup, 2012]. Nesse estudo é apontado no inicio do projeto é comum que se tenha a definição de uma meta, que se pretende cumprir em determinada quantidade de tempo, mas que leva um tempo diferente do planejado inicialmente, isso pode ser dado por fatores diversos, internos ou externos ao desenvolvimento do sistema, situações direta ou indiretamente relacionadas ao sistema.

**Back-End**

Para o desenvolvimento web, back-end é o termo usado para descrever as funções que são executada em um servidor, sem ter que utilizar o processamento da máquina do cliente e retorna um resultado para o navegador utilizado(Tabless, 2015). O back-end é o responsável pelas atuação das funcionalidades do sistema.

**Front-End**

Front-end é pertencente ao DOM (Document Object Model) que nada mais é do que o modelo utilizado para a leitura no navegador, é a base para uma outra árvore que é o que realmente um browser monta na tela. (Tabless, 2011). Ou seja, tudo que é tangível ao usuário, como textos, botões, animações, etc. O desenvolvimento Front-End está diretamente relacionado ao HTML, CSS e JavaScript.

**Framework**

O Framework é baseado em comandos já pré-determinados a fim de que se haja facilidade para o desenvolvimento do sistema, seja qual for. Os frameworks já apresentam funcionalidades prontas, com isso o desenvolvedor tem a facilidade de, com poucas linhas de código, criar rotinas, efeitos, funcionalidades, e outros (de acordo com aquilo a que se proprõe o desenvolvimento do sistema).

**MVC**

MVC (Model, view, controller), está relacionado em fragmentar um projeto em 3 partes, que seriam justamente em “modelo, visão e controlador” (Devmedia). Sendo assim, o usuário primeiramente interage com a Interface (visão), embora quando pressione um botão esteja se relacionando com o controlador, pois este controlador é que faz todo back-end do sistema. Segundo Devmedia, O controlador não conhece diretamente o que é mostrado na View, tão somente envia mensagens quando ela precisa que algo na tela seja atualizado.

**HTML**

O HTML (HyperText Markup Language), é uma linguagem utilizada para desenvolvimento de página WEB e permite a criação que documentos que rodam em diversos tipos de computadores, o qual é exibido no próprio navegador. Este por sua vez é considerada a linguagem base para a internet, foi criada para ser de fácil entendimento por seres humanos (tabless, 2011).

A linguagem é divida em tags, marcando os elementos para mostrar quais informações a página exibe. Por exemplo, quando existe um título importante, normalmento é usada a tag H1, que faz com que o título esteja com um tamanho maior em relação ao convencional.

A estrutura principal de uma pagina html é dividida em: Head e Body (cabeçalho e corpo) dentro da tag <HTML>. Dentro do <BODY> é onde ocorre a parte que será exibida no navegador, e na tag <HEAD> é indicado o título do documento. Nessa tag ainda é declarada a tabela de caracteres que o browser deve usar para renderizar o texto, bem como usar tag que conectem a folhas de estilo (CSS) e scripts em javascript para customizar a página. (tabless, 2011)

**CSS**

O CSS (Cascading Style Sheets) é atualmente a melhor maneira de estilizar a página web. Proporcionando a possibilidade de criar uma página com características próprias, fortemente atrelado ao fato de criar a identidade visual do site. Isso é possível pois esta linguagem modifica o layout da página. Basicamente o CSS proporciona mais opções de construção de layout, a principal diferença entre CSS e HTML é que HTML é usado para estruturar conteúdo e o CSS é usado para formatar conteúdos estruturados. (HTML.NET, 2009)

O CSS foi proposto quando foi notado que desenvolvedores web queriam estilizar suas paginas de diversas formas e isso era feito tag por tag, e linha por linha e foi quando em 1995, Håkon Wium Lie e Bert Bos apresentaram a proposta do CSS. A idéia geral era, utilizar HTML somente para estruturar o website e a tarefa de apresentação fica com o CSS disposto em um arquivo separado com extensão em “.css” ou no próprio HTML demarcado pelas tags.(DEVMEDIA, 2011).

O conceito de CSS não segue um padrão, isso ocorre por conta das incompatibilidades com browsers e também da falta de experiência de alguns desenvolvedores, porém isso permitiu que diversos frameworks (ferramentas de desenvolvimento já prontas) fossem criados para facilitar a estilização e otimização, um dos destaques se da pelo design responsivo que pode adaptar uma pagina a vários tamanhos de tela de diversos dispositivos(DEVMEDIA, 2010).

**Javascript**

JAVASCRIPT que é uma linguagem de programação client-side. Ela é utilizada para controlar o HTML e o CSS para manipular comportamentos na página. Um comportamento comum, por exemplo, é um submenu. efeitos que podem ser vistos em paginas web hoje em dia, como o fadeIn (que faz surgir um elemento no DOM), quando se passa o mouse por cima de um menu janelas são ativadas, entre outros efeitos e recursos. (Tabless, 2015).

JAVASCRIPT é uma linguagem orientada a objetos (onde os elementos e métodos de funções são vistos como objeto, algo que possua caracteristicas, objeto é um grupo de dados), e dentro dela existe o JQUERY, que garante que possa se utilizar e programar diversos efeitos na pagina HTML com poucas linhas de comando se comparado com JAVASCRIPT, o JQUERY é uma biblioteca de JAVASCRIPT simplificada porem poderosa, abrindo inúmeras possibilidades para desenvolvimento web.(TABLESS, 2015)

**IDE**

IDE significa Ambiente Integrado de Desenvolvimento, do inglês Integrated Development Environment (Intellectuale, 2012). Esses ambientes de desenvolvimento proporcionam integração de diversos recursos para facilidade do desenvolvedor, seja isso o próprio compilador ou plugins que podem ser instalados. Assim como corretor léxico e sintático do código em questão.

**PHP**

PHP é uma linguagem de script open source de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML.(PHP.NET, 2009). Essa é uma linguagem já muito usada no mercado web a anos, com diversas ferramentas e IDE’s de fácil uso para poder aproveitar o poder máximo da linguagem. (Dicas de Programação, 2014).

Suas principais vantagens são: linguagem dinâmica ou seja é necessário declarar tipos e variáveis para trabalhar com valores, em uma mesma variável pode existir números e ou texto(PHP.net, 2009), onde, no Course Factory SYS ela realiza uma fundamental tarefa de controle, capturando os dados da página do cliente e enviando ao servidor PHP onde o Back-End realiza as operações funcionais da página.

Cliente e servidor é a maneira de como é feita a troca de informações pela rede, através de mensagem e resposta. O “Cliente” são as operações que acontecem no navegador do usuário, enquanto o servidor é o local que realiza operações também, porém de forma segura e onde o cliente envia mensagens para o servidor poder processar informações e mostrar aos usuários no navegador. (Tabless, 2015) Hoje em dia muitos sistemas já utilizam o modelo peer-to-peer onde o cliente pode ser tanto servidor como também cliente e vice-versa. (Kioskea, 2015)

**Banco de Dados**

Segundo Furtado, 2013, Bancos de dados são coleções de dados que se relacionam e são organizados de modo a realizar a persistência de informações, assim, sendo possível realizar o CRUD das informações do sistema.

Nesse ponto é de suma importância ressaltar a diferença entre Dados e Informações:

* Dados: Fatos em sua forma primária. Em sua maioria, dados sozinhos não fazem sentido.
* Informações: Agrupamento de dados de modo a gerar coesão e sentido para que se possa gerar conhecimento.

**Design FLAT**

O design FLAT é caracterizado por ser “clean” e objetivo, sem muitos efeitos de cores ou qualquer outra característica que possa poluir e causar interferência visual na mensagem que se deseja passar.

Segundo SHINMI, 2014 o design Flat tem sido conhecido como “Design Plano”, tem suas figuras minimalistas seguindo o ideal de que “mais é menos”, e na presente data tem sido a grande tendência em design, por visar principalmente a usabilidade, sendo assim, o design tem cada vez mais focado no usuário.

**Ferramentas utilizadas**

A primeira ferramenta usada para o desenvolvimento do sistema foi a IDE PHPstorm da JetBrains, para PHP, ambiente que também é possibilita editar e criar códigos em HTML, CSS, JAVASCRIPT. Esta IDE possui um “debugger” que é um analizador de código que faz o passo a passo do código, onde o usuário pode verificar as etapas e processos de seu código. A escolha dessa ferramente foi feita partindo do ponto em que, além dos recursos que facilitam a programação, o PHPStorm possui uma estrutura de entendimento intuitivo e fácil utilização.

Para servidor foi utilizado o XAMPP, este que por sua vez é um servidor independente que proporciona o ambiente para teste necessário para análise da performática do sistema por ter sido desenvolvido com o intúito de proporcionar um depurador PHP e possui ainda o apache.

Para banco de dados foi usado MySQL, pois esse sistema gerenciador de banco de dados trabalha com com a linguagem SQL. O MySQL é utilizado a fim de realizar a persistência de dados. O MySQL é um sistema cliente/servidor multitarefa que tem capacidade de suportar diferentes acessos simultaneamente, ferramentas administrativas e diversas APIs que facilitam seu uso. As principais características do MySQL são: escalabilidade e desempenho, o que proporciona utilização mesmo em sistemas críticos; flexibilidade e compatibilidade com diversas plataformas e linguagens de programação; e possui várias funções para manipulação de dados permitindo um sistema com controle e estabilidade.

SQL é hoje um dos mais utilizados no mercado quando se fala de desenvolvimento de aplicações Web, pela sua facilidade e excelente curva de aprendizado tornando ela prioridade de escolha na hora de iniciar um projeto(TechTudo, 2010)

Foi desenvolvido pela empresa sueca MYSQL AB e publicada em 1995, pouco tempo depois a mesma foi comprada pela Sun Microsystems que foi comprada atualmente pela Oracle que cuida do SQL.(TechTudo, 2010).

Uma das grandes facilidades e principal motivo de usar MySQL é sua integração com PHP que é extremamente simples e existem diversas ferramentas no PHP para a manipulação do SQL, alem de exigir bem pouco do hardware, suporta trigges (gatilhos de evento dentro do MySQL, quando uma função é executada, outro comando é disparado), possui um desempenho incrível e também é multitarefa.

Neste projeto foi utilizado um framework chamado Semantic-UI para desenvolvimento e definição do layout, este que por sua vez possui a folha de estilo pronta para ser trabalhada com efeitos que é uma classe CSS pronta com estilizações prontas (Semantic-UI, 2014) em JQuery (Javascript) o que facilita o desenvolvimentos de funções para a página funcionando no client-side, (DEVMEDIA, 2011).

O layout foi baseada na identidade visual do atual sistema do Cliente, com a cor vermelha escura com um fundo cinza. O framework Semantic-UI possibilita uma estrutura compatível ao requerido pelo Cliente, onde as telas ficaram com a característica Flat, bastante atrativo pois ficou um ambiente bastante “clean”.

**Desenvolvimento**

O desenvolvimento do projeto partiu do ponto da escolha do tema, que foi dado a partir de uma sugestão do professor instrutor do grupo, Marcos Reis. Então o grupo analisou os pontos cruciais de desenvolvimento junto ao Cliente e definiu que seria desenvolvido o Sistema de Oferta de Disciplinas, denominado de *Course Factory SYS.*

Para desenvolvimento do sistema CFS, o grupo realizou uma reunião inicial com o Cliente, para alinhar quais seriam os requisitos do sistema. Com o reconhecimento do Stakeholder, foi usada a metodologia de desenvolvimento ágil chamada SCRUM, para determinar quais seriam os tópicos a serem abordados inicialmente com a definição do backlog.

Dentro do método do SCRUM, caracterizamos o Cliente como Product Owner, o professor instrutor do grupo, Marcos Reis como SCRUM MASTER e o restante do grupo como o Time de Desenvolvimento.

Sabendo a magnitude do projeto, o grupo acordou em fragmentar o projeto em duas etapas de desenvolvimento:

|  |  |
| --- | --- |
| **Fase** | **Descrição** |
| Etapa 1 | * Backlog * Roadmap * Definição de layout * CRUD Curso * CRUD Turma * CRUD Disciplina * CRUD Aluno |
| Etapa 2 | * Analisador de disciplina * Gerador de painel de ofertas * Manipulação dos resultados do painel * Melhorias da Etapa 1 (se necessário) * Geração de relatório * Homologação |

Tabela 3.0 - Etapas do projeto

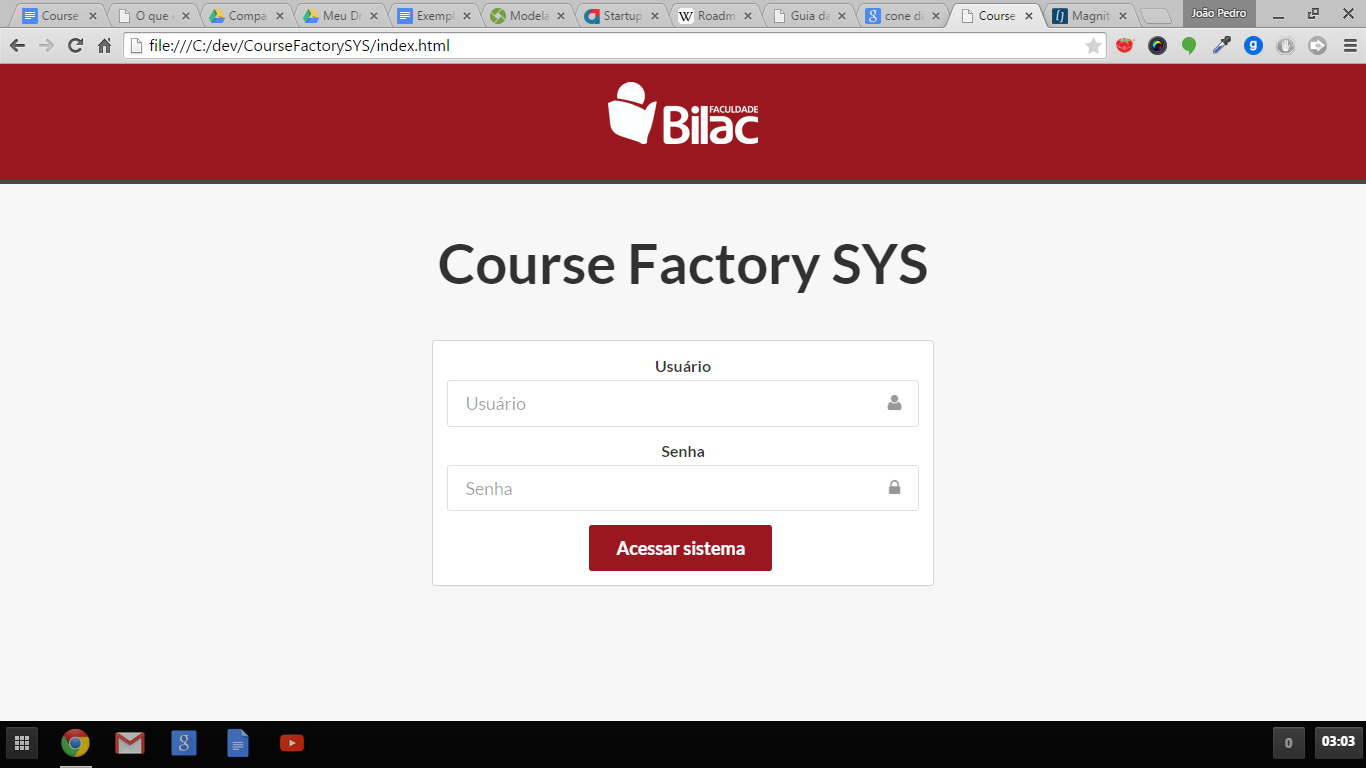
A primeira etapa do projeto, foi definida como tendo o término ao final do primeiro semestre de 2015.

Para que o grupo tivesse compreensão dos prazos, e para que se tivesse uma visão de como realizar os Sprints, junto ao SCRUM MASTER, o grupo optou em definir um Roadmap, logo após ter feito a definição do backlog.

Para desenvolver o layout, foi usado inicialmente o editor de texto SUBLIME-Text, que foi logo substituído pelo PHPStorm, ferramenta que foi escolhida justamente pela sua eficiência para auxiliar no desenvolvimento de sistemas. O layout foi desenvolvido com base nas cores do próprio sistema do Cliente, mantendo a identidade visual da instituição utilizadora do sistema, Faculdade Bilac.

O layout foi desenvolvido utilizando CSS e para poder usufruir de recursos personalizados e mais atrativos ao usuário, foi utilizado o *Framework* Semantic-UI.

Para exemplo de layout, segue abaixo a imagem da tela de login, esta que é uma das principais telas, pois através da autenticação do usuário, garantindo a segurança do sistema.

Figura 2.0 - Tela de login

Nessa tela o usuário deve entrar com seus dados já cadastrados previamente, sendo eles: Usuário e Senha. Após preenchido os dados e pressionado o botão, ali destacado em vermelho com a mensagem “Acessar sistema”, o sistema busca no banco de dados se as informações ali preenchidas estão de acordo com o que há cadastrado. Caso a informação seja autenticada (válida) o usuário terá acesso ao sistema, caso contrário, (a não validação das informações preenchidas) será enviada uma mensagem ao usuário, informando que os dados não são válidos e que ele deve verificar.

Essa foi a primeira tela solicitada pelo SCRUM MASTER, porém, originalmente ela havia somente um botão de acesso, e no segundo Sprint foi solicitado que houvesse o *upgrade* para que assim fosse preservada a segurança do sistema.

As funcionalidades *back-end* foram desenvolvidas utilizando PHP, linguagem escolhida devido a facilidade encontrada pelos membros do grupo em realizar as manipulações de dados, fácil aprendizado e prévio conhecimento da linguagem por parte de um dos membros do grupo. Para o desenvolvimento nessa linguagem também foi utilizada a IDE PHPStorm.

No primeiro Sprint foi definido que deveria ser desenvolvida a tela de login, como citado acima, também de que seria necessário criar o banco de dados, o qual o grupo optou em utilizar do serviço oferecido pelo MySQL para fazer a persistência de dados, uma vez que os membros do grupo já tiveram um contato anterior com o ambiente. Outro fator de grande valia no momento da escolha da ferramente é o fato de essa ser bastante difundida em diversas áreas das computação, (usada por muitos programadores) e com isso é facilmente encontrado na rede grande quantidade de suporte e recursos que facilitam para os desenvolvedores garantirem a performance e qualidade, e ainda por ser esta uma ferramenta gratuita. Ainda no primeiro Sprint, foi definido que o Time (grupo do *Course Factory SYS*) deveria desenvolver as primeiras característas dos CRUD’s, e foi colocado no Sprint que o CRUD inicial seria o das disciplinas.

Abaixo são demonstradas algumas telas referente ao desenvolvimento do proposto:

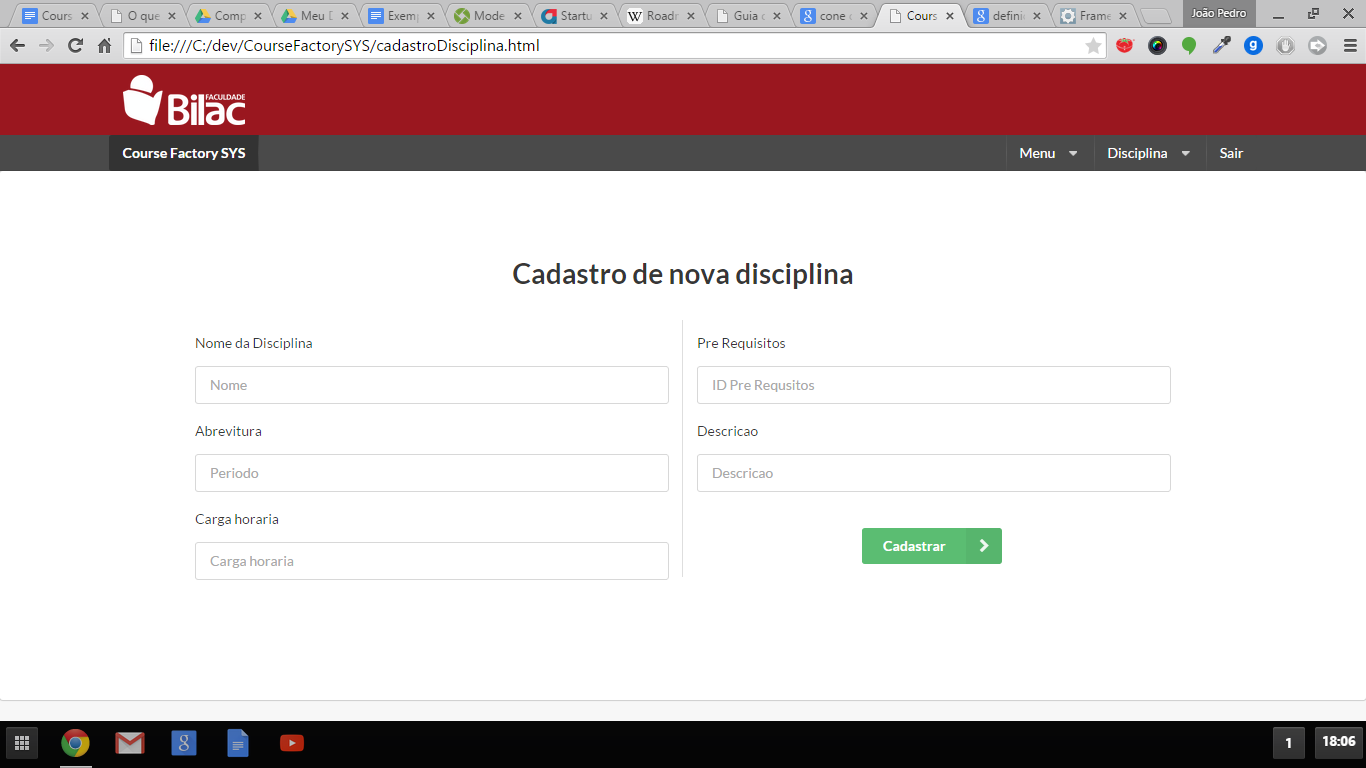


Figura 3.0 - Cadastro disciplina

Nessa tela, é onde o usuário deverá preencher as informações para cadastrar uma nova disciplina. Todas informações são obrigatórias, sendo elas:

* Nome da disciplina
* Abreviatura
* Carga Horária
* Pré-requisitos\*
* Descrição

(\*os pré-requisitos na disciplina foi acordado depois, junto ao SCRUM Master, de que deveria ser em uma relação da disciplina com o curso, pois uma mesma matéria, em cursos diferentes pode ter, ou não, pré-requisitos diferentes).

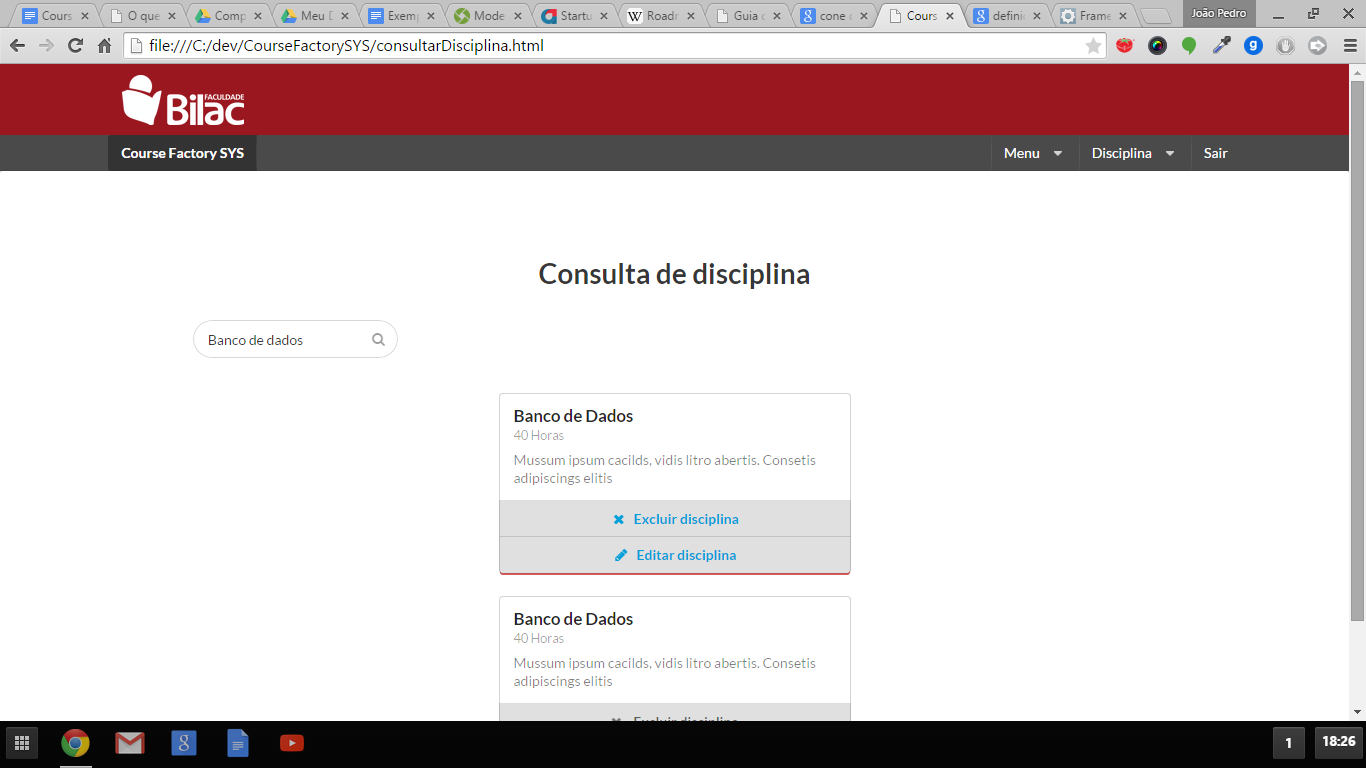


Figura 4.0 - Consultar disciplinas

Ao consultar uma disciplina, o usuário já tem a possibilidade de selecionar a opção de alterar ou mesmo de excluir aquela determinada disciplina. Isso foi feito já com o intuito de melhorar a usabilidade do sistema. Para essa etapa, na hora da busca o sistema realiza uma busca, no banco de dados, quais as disciplinas tem o nome similar ao que foi digitado para busca.

Feito isso, as telas foram mostradas para o SCRUM Master, junto do Product Owner, que aprovaram o layout e a maneira e a maneira como o sistema deveria reagir àquelas ações.

O Time então já com a base de como “desenhar” as próximas telas e realizar os próximos passos, fechou o segundo Sprint, junto com o SCRUM Master, onde ficou determinado de seria necessário a elaboração do CRUD do Curso, o que demandou um pouco mais de tempo do Time, pois houveram algumas dificuldades na integração de todos os recursos entre todos membros do time.

Superadas as dificuldades encontradas de integração, foi dado prosseguimento do projeto onde foram desenvolvidas as telas do CRUD de Curso. Abaixo segue duas figuras da tela de Cadastro do Curso.

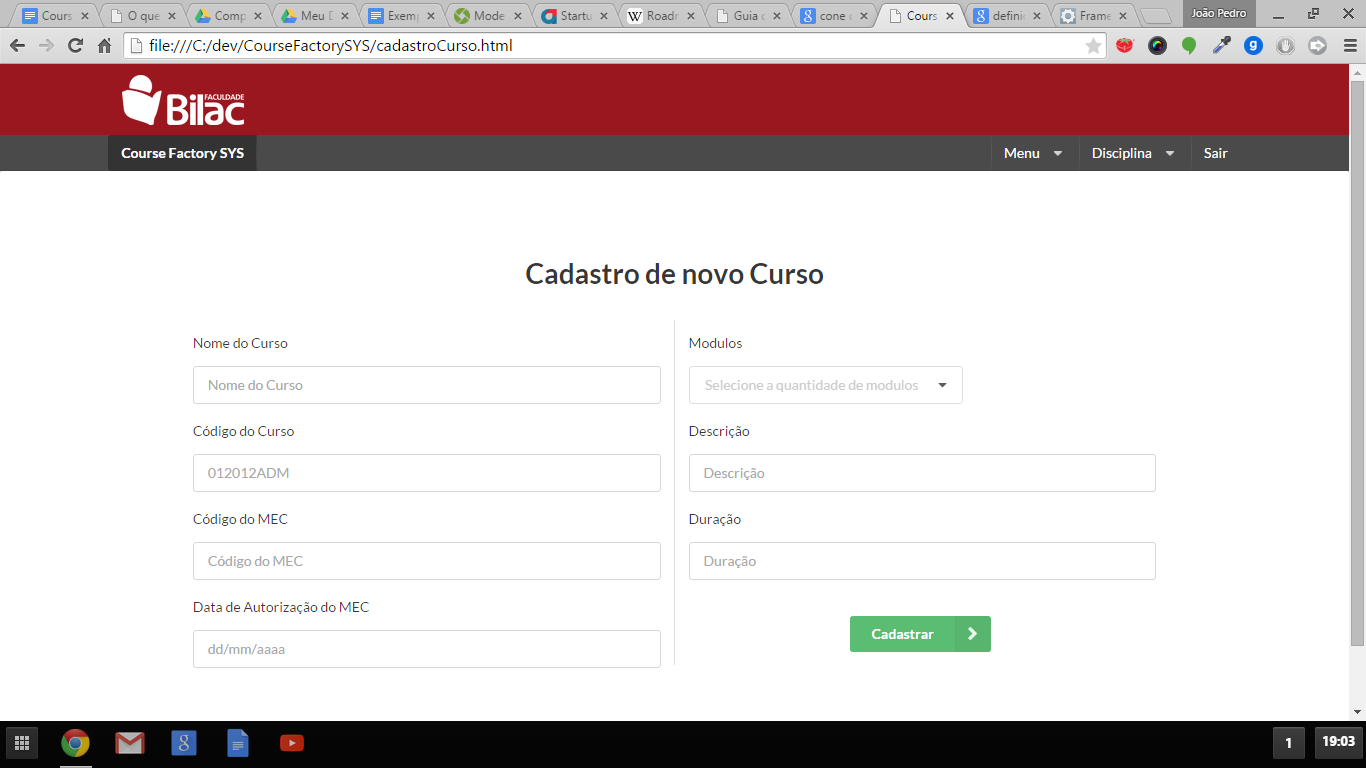


Figura 5.0 - Cadastro de novo Curso

Nessa etapa o usuário deve preencher todas informações para que seja cadastro o novo curso. Os campos obrigatórios para o cadastro do curso são:

* Nome do Curso
* Código do Curso
* Código do MEC
* Data de autorização do MEC
* Quantidade de módulos
* Coordenador
* Duração do curso

Logo após a criação da tela de cadastro do curso, o Time criou as telas de Alterar e Consultar Curso, onde, nas duas figuras a seguir são expostas as propostas do sistema ao que compreende a Alteração do Curso.

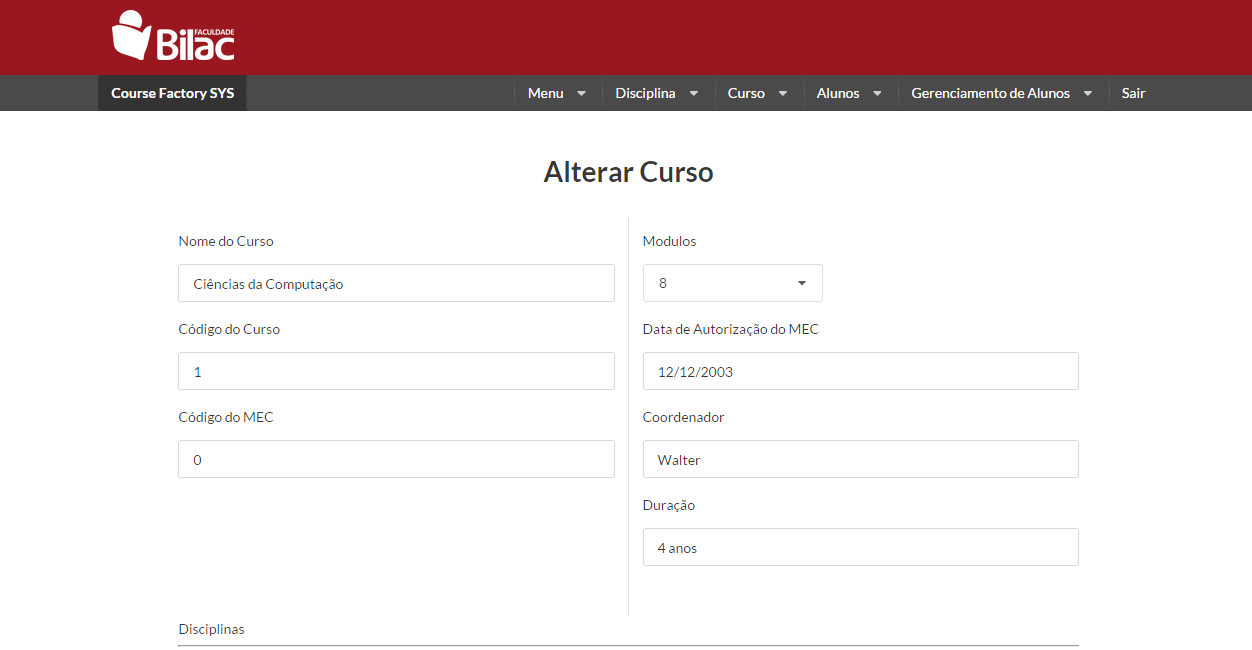


Figura 6.0 - Alterar curso

A parte superior da tela de alterar curso é bastante parecida com a do cadastro, para que seja mantida a mesma concepção do layout, e disposição dos campos e assim o aprendizado seja de maneira bastante intuitiva.

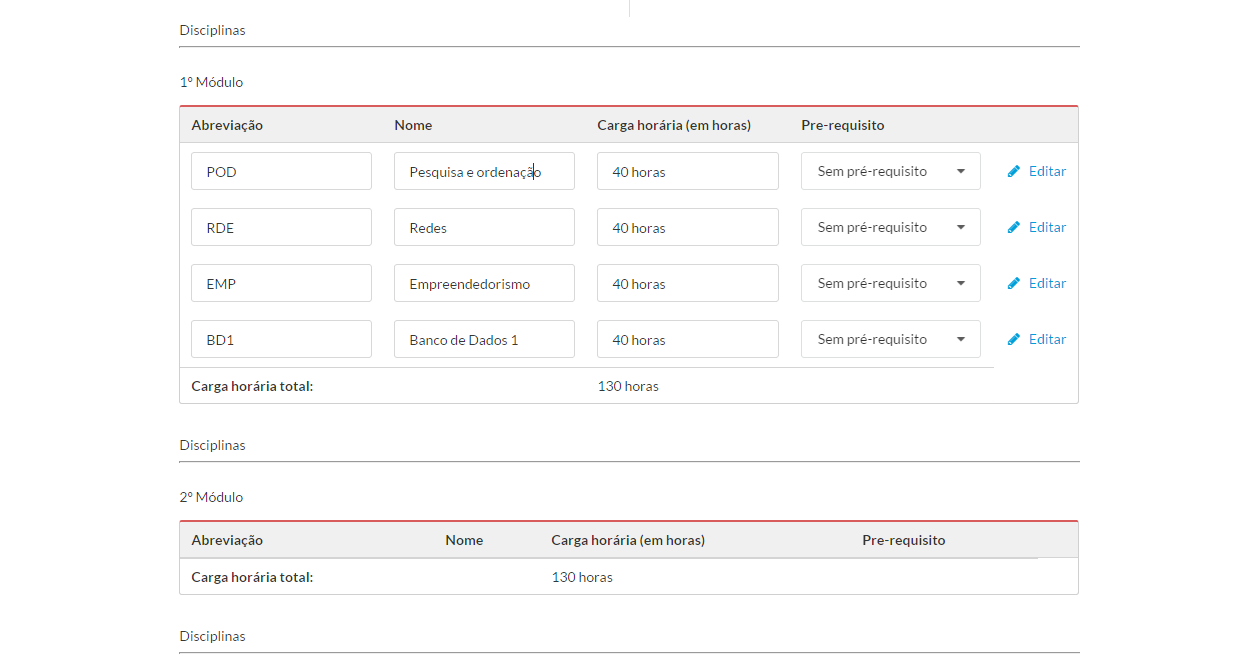


Figura 6.1 - Alterar curso

A segunda parte da tela é onde apresenta os módulos daquele curso e quais são as disciplinas relacionadas a ele. Nesse ponto, caso seja realizado uma alteração na disciplina, essa alteração será relacionada à disciplina, com exceção do campo de “Pré-Requisito” que é uma relação especial entre o Curso e aquela determinada Disciplina.

No período do desenvolvimento do CRUD de Curso foi justamente o período de semana de provas, inclusive teve que fazer uma apresentação do sistema para o SCRUM Master junto do Product Owner (o qual não pode estar presente), e por esse motivo o Time teve uma pequena queda na produção, porém o Time manteve-se focado e conseguiu concluir essa etapa com êxito.

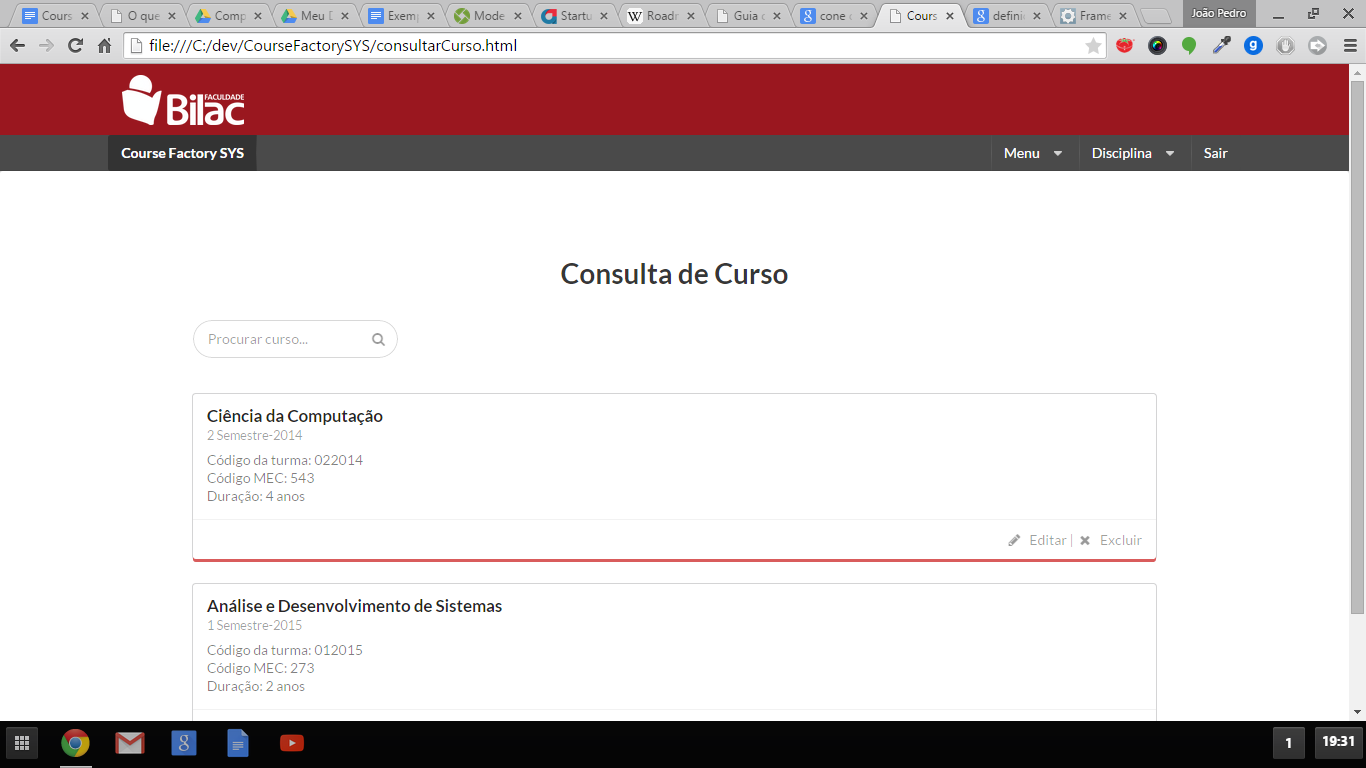


Figura 7.0 - Consulta de Curso

Nessa etapa o usuário pode realizar a busca pelo curso o qual deseja consultar alguma informação, alterar ou excluir.

Finalizando o CRUD do Curso o foco voltou-se a desenvolver os CRUD’s de Aluno e Turma, o que foi facilitado, porque o grupo estava usando a técnica de MVC.

Após ser criado o CRUD de ambas, notou-se a necessidade de criar a relação direta do Aluno com a Disciplina, afinal através dessa é que se daria para realizar a avaliação necessária da Etapa II do projeto. Com isso, teve em última instância do projeto a criação da relação Aluno x Disciplina, como última fase de sua Etapa I.

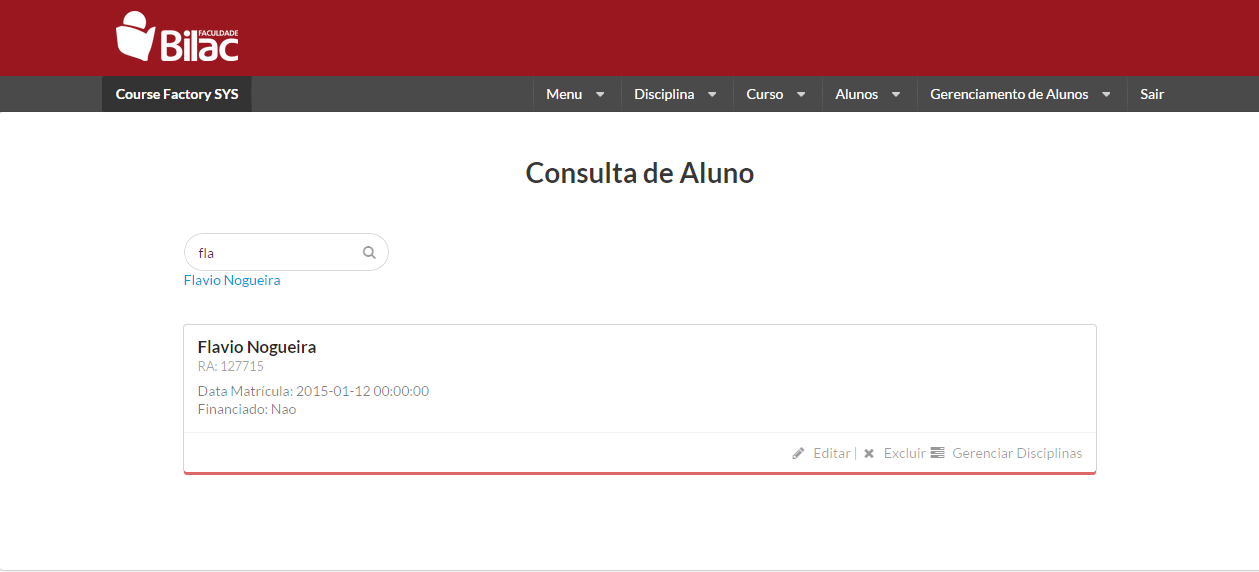


Figura 8.0 - Consulta aluno

Assim quando um aluno é buscado, pode-se escolher a opção de “Gerenciar Disciplinas” onde o usuário será levado para a página das “disciplinas do aluno”, exemplificada abaixo.



Figura 9.0 - Disciplinas do Aluno

Em vista do desenvolvido, com a conclusão dessa Etapa, o Time partir desse ponto está apto a prosseguir com a Etapa II do projeto.

# Referências

Guia Pronatec - Curso superior. Disponível em:<http://www.guiapronatec.com.br/pronatec-diferencas/>

Acesso em 15 de maio de 2015

Curso Superior de Tecnologia - O que é isso? Disponível em: <http://www.utfpr.edu.br/apucarana/cursos/tecnologias/Ofertados-neste-Campus/curso-superior-de-tecnologia-o-que-e-isso>

Acesso em 15 de maio de 2015

CURSO TÉCNICO, TECNOLÓGICO OU SUPERIOR, Disponível em:<http://reitoria.ifpr.edu.br/menu-institucional/acesso-a-informacao/perguntas-frequentes/curso-tecnico-tecnologico-ou-superior/>

Acesso em 10 de maio de 2015

FURTADO, G. As 10 linguagens de programação mais requisitadas pelo mercado. Disponível em: <http://www.dicasdeprogramacao.com.br/as-10-linguagens-de-programacao-mais-requisitadas-pelo-mercado/>

Acesso em 19 de maio de 2015

2014, O que é o PHP? Disponível em:

<http://php.net/manual/pt_BR/intro-whatis.php>

Acesso em 20 de maio de 2015

PISA, P - O que é e como usar o MySQL? Disponível em: <http://www.techtudo.com.br/artigos/noticia/2012/04/o-que-e-e-como-usar-o-mysql.html>

Acesso em 18 de maio de 2015

Tableless. O que é client-side e server-side? Disponível em:<http://tableless.github.io/iniciantes/manual/obasico/o-que-front-back.html>

Acesso em 01 de junho de 2015

PINTUDA, 2015 - O que é o Peer-to-peer (P2P) ? Disponível em: <http://pt.kioskea.net/faq/10017-o-que-e-o-peer-to-peer-p2p>

Acesso em 19 de maio de 2015

EIS, D - O básico: O que é HTML? Disponível em:

<http://tableless.com.br/o-que-html-basico/>

Acesso em 15 de maio de 2015

Lição 1: O que é CSS? Disponível em

[ttp://pt-br.html.net/tutorials/css/lesson1.php](http://pt-br.html.net/tutorials/css/lesson1.php)

Acesso em 20 de maio de 2015

O que é jQuery? Disponível em: <http://tableless.github.io/iniciantes/manual/js/o-que-jquery.html>

Acesso em 20 de maio de 2015

CASAVELLA, E. Ambientes Integrados de Desenvolvimento em Linguagem C, 2012. Disponível em: <http://linguagemc.com.br/ambientes-integrados-de-desenvolvimento-em-linguagem-c/>

Acesso em 24 de maio de 2015.

SILVA, J. Análise e Especificações de Requisitos, Disponível em: <http://www.linhadecodigo.com.br/artigo/3224/analise-e-especificacoes-de-requisitos.aspx>

Acesso em 31 de maio de 2015.

PRESSMAN, Roger, S. Engenharia de Software. 6. ed. São Paulo: Bookman, 2006.

SOMMEVILLLE, Ian. Engenharia de Software. 8. ed. São Paulo: Pearson, 2008.

SCHWABER, K. Guia do SCRUM, 2009. Disponível em: <http://www.trainning.com.br/download/GUIA_DO_SCRUM.pdf>

Acesso em 31 de maio de 2015.

Oracle Brasil, MySQL. Disponível em: <http://www.oracle.com/br/products/mysql/index.html>

Acesso em 29 de maio de 2015

Jetbrains, Instalando e configurando o XAMPP com o PhpStorm. Disponível em: <https://confluence.jetbrains.com/display/PhpStorm/Instalando+e+configurando+o+XAMPP+com+o+PhpStorm>

Acesso em 30 de maio de 2015

DEVMEDIA, Visão Geral do Sistema de Gerenciamento de Banco de Dados MySQL. Disponível em: <http://www.devmedia.com.br/visao-geral-do-sistema-de-gerenciamento-de-banco-de-dados-mysql/2981>

Acesso em 31 de maio de 2015

DIAS, S - Gestão de Projetos e Software. Disponível em: <http://www.sergiodias.inf.br/engenharia-de-software/gestao-de-projetos-de-software>

Acesso em 30 de maio 2015

RODRIGUES, N - Modelagem de Negócio - Engenharia de Software 31. Disponível em: <http://www.devmedia.com.br/modelagem-de-negocio-engenharia-de-software-31/18729#ixzz3c9R8Th2a>

Acesso em 01 de junho de 2015

JACQUES - Descrição dos Requisitos Não Funcionais. Disponível em: <http://www.dsc.ufcg.edu.br/~jacques/cursos/proj/gerenciadesenv/naofuncionais.htm>

Acesso em 04 de junho de 2015

UNIVEL - Graduação. Disponível em:

<http://www.univel.br/graduacao.php>

Acesso em 07 de setembro de 2015

Mobgeek, 2014. O que é Back-end como serviço (Backend as a Service)? Disponível em: <http://mobgeek.com.br/blog/o-que-e-back-end>

Acesso em 05 de junho de 2015

SHINMI, M, 2014. O conceito Flat Design. Disponível em: <http://www.oxigenweb.com.br/artigos/flat-design-o-que-e-de-onde-vem/>

Acesso em 05 de junho de 2015

PORTO, Gabriela - Educação - InfoEscola. Disponível em:

<http://www.infoescola.com/educacao/diferencas-entre-tecnologo-bacharelado-e-licenciatura/>

Acesso em 07 de setembro de 2015.

OLIVEIRA, Julio, 2009. Saiba como funciona sistema de ensino superior no Brasil. Disponível em:

<http://www.brasil.gov.br/educacao/2009/11/ensino-superior>

Acesso em 09 de setembro de 2015