



UNIVERSIDADE FEDERAL DO ABC
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ESTÁGIO SUPERVISIONADO

ESTÁGIO EM TECNOLOGIA DA INFORMAÇÃO NA EMPRESA TRIPLETECH IT SOLUTIONS

LUCAS GABRIEL TEIXEIRA

Orientador: Prof. Dr. Edson Pinheiro Pimentel

Santo André, 2015

LUCAS GABRIEL TEIXEIRA

**ESTÁGIO EM TECNOLOGIA DA INFORMAÇÃO NA
EMPRESA TRIPLETECH IT SOLUTIONS**

Trabalho apresentado à banca de Estágio
Supervisionado como requisito parcial para
obtenção do título de Bacharel em Ciência da
Computação pela Universidade Federal do
ABC.

Professor: Prof. Dr. Edson Pinheiro Pimentel

Santo André, 2015

DEDICATÓRIA

Dedico todo projeto realizado aos meus amigos e familiares que depositaram grande confiança em mim desde o início.

AGRADECIMENTOS

Meus agradecimentos são voltados à empresa Tripletech IT Solutions, principalmente ao setor de Desenvolvimento em Tecnologia da Informação, que propôs essa oportunidade de amadurecer meus conhecimentos obtidos na Universidade Federal do ABC, além do crescimento pessoal.

EPÍGRAFE

“A maioria das pessoas pensa no sucesso e no fracasso como opostos, mas eles são ambos produtos do mesmo processo.”

Roger Von Oech

RESUMO

O estágio consiste na manutenção de sistemas existentes em PHP, tanto nas permissões de acesso de usuários quanto segurança das informações. Geralmente são correções e adição de informações nas funcionalidades já existentes. Novos sistemas são desenvolvidos com o framework .NET, banco de dados SQL Server e padrão MVC. A fim de melhorar a experiência do desenvolvimento e do usuário final, ferramentas modernas são exploradas, como o AngularJS by Google. O sistema de bolsa de estudos foi o primeiro com a ferramenta e, após êxito no desenvolvimento e nos testes, foi iniciado o sistema de gerenciamento de projetos com exploração máxima dos recursos, com mais agilidade e padrão no desenvolvimento e uma experiência mais ativa e dinâmica por parte do usuário final.

ABSTRACT

The internship is the maintenance of existing systems in PHP, both on user access permissions as information security. Usually fixes and adding information to existing functionality. New systems are developed with the .NET framework, SQL Server database and MVC pattern. To enhance the experience of development and user, modern tools are explored, such as angularjs by Google. The scholarship system was the first with the tool and, after successful development and testing, started the project management system with maximum exploitation of resources, more quickly and standard development and a more active and dynamic experience by the user.

LISTA DE ILUSTRAÇÕES

Figura 2.1 - Ordenação das palestras por data.	15
Figura 2.2 - Bloqueio de atualização de eventos após a sua data de acontecimento.	15
Figura 2.3 - Tela inicial de Acesso com o campo de matrícula.	16
Figura 2.4 - Confirmação de um dos telefones cadastrados e e-mail em Acesso.	17
Figura 2.5 - Formulário antigo do PROMAIS em Microsoft Access.	19
Figura 2.6 - Diagrama de relacionamento do banco de dados do PROMAIS.	21
Figura 2.7 - Tela de login da aplicação PROMAIS.	22
Figura 2.8 - Tela inicial da aplicação PROMAIS.	22
Figura 2.9 - Tela de alteração ou criação de registro.	23
Figura 2.10 - Informações do edital em destaque.	24
Figura 2.11 - Tela de resumo e prévia de impressão.	24
Figura 2.12 - Tela inicial do Runrun.it [http://bit.ly/1OgVeKa]	26
Figura 2.13 - Visualização de site responsivo em diversas resoluções de monitor [http://bit.ly/1FgliiF].	26
Figura 2.14 - Diagrama de relacionamento do banco de dados do Sistema de Gestão de Atividades.	28
Figura 2.15 - Tela inicial do sistema.	29
Figura 2.16 - Dashboard do sistema.	30
Figura 2.17 - Antiga tela de exibição dos detalhes, em abas, da tarefa quando expandida e filtros de status na barra lateral direita.	31
Figura 2.18 - Tela principal com as lista de atividades criadas em visualização de lista.	32

Figura 2.19 - Antiga tela principal com a opção de “nova tarefa” aberta e filtros de status na barra lateral direita.	33
Figura 2.20 - Tela para edição de tarefa com visualização de comentários à direita.	34
Figura 2.21 - Tela para criação de projeto.	35
Figura 2.22 - Visualização de rascunho de projetos.	36
Figura 2.23 - Criação de Reunião.	37
Figura 2.24 - Visualização de criação de plano de trabalho.	38
Figura 3.1 - Estrutura do HTML com novas tags introduzidas na versão 5.	43
Figura 3.2 - Estrutura de uma árvore DOM.	44
Figura 3.3 - Exemplo de manipulação de DOM com Javascript.	44
Figura 3.4 - Exemplo de criação de nó com AngularJS.	45
Figura 3.5 - Métodos de utilização do CSS [http://bit.ly/1G00zSp].	46
Figura 3.6 - Exemplo de conexão básica com o bando de dados em C#.	47
Figura 3.7 - Expressão Regular para telefone.	48
Figura 3.8 - Modelo de consulta em banco de dados.	49

LISTA DE ABREVIACÕES E SIGLAS

CRUD	Create, Read, Update and Delete
DOM	Document Object Model
JSON	JavaScript Object Notation
MVC	Padrão de arquitetura de software
MySQL	Sistema de gerenciamento de banco de dados
.NET	Plataforma de desenvolvimento
SQL	Structured Query Language (Linguagem de Consulta Estruturada)
SQL Server	Sistema de gerenciamento de banco de dados
TI	Tecnologia da Informação
UFABC	Universidade Federal do ABC

SUMÁRIO

1.	INTRODUÇÃO.....	11
1.1.	CARACTERÍSTICAS DA ÁREA DO ESTÁGIO.....	11
1.2.	CARACTERIZAÇÃO E ANÁLISE DA EMPRESA	11
1.3.	VISÃO GERAL DO ESTÁGIO	12
1.4.	ORGANIZAÇÃO DO TRABALHO.....	13
2.	ATIVIDADES DESENVOLVIDAS.....	14
2.1.	MANUTENÇÃO DE SISTEMAS EM PHP	14
2.1.1.	RESULTADOS	17
2.1.2.	CONCLUSÃO.....	18
2.2.	DESENVOLVIMENTO DO SISTEMA DE BOLSA DE ESTUDOS PROMAIS.....	18
2.2.1.	RESULTADOS	20
2.2.2.	CONCLUSÃO.....	25
2.3.	DESENVOLVIMENTO DO SISTEMA DE GESTÃO DE ATIVIDADES	25
2.3.1.	RESULTADOS	28
2.3.2.	CONCLUSÃO.....	38
2.4.	DIAGNÓSTICO DOS PRINCIPAIS PROBLEMAS OBSERVADOS E SUGESTÕES DE MELHORIA.....	39
3.	FUNDAMENTAÇÃO TEÓRICA.....	41
3.1.	PLANEJAMENTO DE PROJETOS	41
3.2.	DESENVOLVIMENTO WEB	42
3.2.1.	HTML.....	42
3.2.2.	JAVASCRIPT	43
3.2.3.	ANGULARJS.....	45
3.2.4.	CSS	45

	10
3.3. LÓGICA DE PROGRAMAÇÃO	46
3.3.1. CONEXÃO COM BANCO DE DADOS	46
3.3.2. EXPRESSÕES REGULARES	47
3.4. BANCO DE DADOS	48
4. CONSIDERAÇÕES FINAIS	50
4.1. CONTRIBUIÇÕES PARA A FORMAÇÃO	50
4.2. DIFICULDADES ENCONTRADAS	50
4.3. SUGESTÕES DE TRABALHOS FUTUROS	51
REFERÊNCIAS BIBLIOGRÁFICAS	52

1. INTRODUÇÃO

Este documento tem por objetivo descrever as atividades e os produtos obtidos na realização do Estágio Supervisionado. Neste capítulo, apresenta-se a caracterização do estágio, da empresa, uma visão geral do estágio e a organização geral do documento.

1.1. CARACTERÍSTICAS DA ÁREA DO ESTÁGIO

O local de trabalho é no setor de Tecnologia da Informação do cliente Faculdade de Direito de São Bernardo do Campo [2], na região do Grande ABC Paulista, mais precisamente no setor de Desenvolvimento. Jornada de trabalho de segunda à sexta-feira das 07h00 às 16h00 com 1 hora de almoço, totalizando 40 horas semanais, com salário de R\$ 2.000,00 por mês.

O setor possui em média 18 funcionários, sendo divididos em Desenvolvimento, Redes, Suporte e funcionários da própria faculdade. Cada funcionário possui sua própria mesa de trabalho e seu próprio computador para realizar suas aplicações e projetos. A área total do setor é suficiente para a quantidade de pessoas que nele trabalham.

O setor é responsável por desenvolver aplicações que facilitem as tarefas dos usuários no dia-a-dia como consulta de dados, cadastro e alteração de informações, relatórios, além de suporte a usuários por telefone ou e-mail relacionados a assuntos diversos como suporte às próprias aplicações criadas.

1.2. CARACTERIZAÇÃO E ANÁLISE DA EMPRESA

O estágio foi iniciado no dia 02/02/2015, contratado pela empresa Tripletech It Solutions Soluções em Ti Ltda - Epp, CNPJ: 09.554.503/0001-83, que atua com suporte técnico, manutenção e outros serviços em tecnologia da informação, está localizada na Av Nazare, 1139, Sala 1106, Ipiranga, Sao Paulo - SP, CEP 04262-100, telefone: (11) 2533-9772.

A Tripletech IT Solutions é uma empresa fundada em 2008 por profissionais de alta qualificação técnica e de negócios, voltada ao fornecimento de soluções de infraestrutura de TI para os mais diversos segmentos [1].

A Tripletech IT Solutions atua no mercado corporativo sempre em busca da excelência dos serviços prestados na área de Tecnologia da Informação, destacando-se nesse segmento por acompanhar as constantes evoluções e tendências de mercado. A empresa é formada por profissionais com ampla expertise adquirida ao longo dos anos no mercado de TI.

As principais certificações da empresa para atuação em projetos são:

- Redes de computadores;
- Colaboração;
- Banco de dados;
- Cloud computing;
- Segurança da informação;
- Virtualização de servidores;
- Outsourcing;
- Migrações;
- Movimentação de datacenter.

Além de capacidade de agregação e gerenciamento de recursos necessários ao desenvolvimento de projetos de pequeno a grande porte.

Na ótica da Tripletech IT Solutions, as necessidades dos clientes estão acima de qualquer interesse e, independente dos parceiros e brands que é comercializada, o compromisso é de total imparcialidade ao analisar um ambiente de TI e propor soluções e melhorias.

O foco é sempre ofertar a melhor solução de acordo com a necessidade na medida certa para cada tipo de ambiente em cada cliente, tratando a TI de uma forma ampla, objetiva e sem complicações [1].

1.3. VISÃO GERAL DO ESTÁGIO

Inicialmente, durante as três primeiras semanas, o estagiário foi introduzido à manutenções em sistemas PHP para familiarização com o ambiente de trabalho e conhecimento das bases de dados. As manutenções consistiam em apresentação de informações com consultas específicas no banco de dados e correção de falhas.

Após o período de integração, foi iniciado o principal projeto do estágio: Sistema de Gestão de Atividades. Esse projeto possui várias características de um grande sistema e exigiu conhecimentos acadêmicos específicos, como: banco de dados, orientação à objetos, programação para web, planejamento e estruturação.

Após um mês de desenvolvimento do Sistema de Gestão de Atividades, foi iniciado o projeto Sistema de Bolsa de Estudos PROMAIS, a fim de testar novas ferramentas que poderiam ser utilizadas no primeiro citado. O Sistema de Bolsa de Estudos PROMAIS demorou dois meses para ser finalizado e então foi iniciado novamente o Sistema de Gestão de Atividades, dessa vez com as novas ferramentas testadas e introduzidas. O projeto permanece em desenvolvimento durante a elaboração deste relatório.

1.4. ORGANIZAÇÃO DO TRABALHO

As próximas seções estão organizadas como segue:

A seção 2 apresenta as informações detalhadas de todas as atividades realizadas pelo estagiário. A seção está dividida em três partes, uma para cada atividade desenvolvida.

A seção 3 descreve as considerações finais em relação a todo o estágio, como: contribuições para a formação, dificuldades encontradas e sugestões para trabalhos futuros.

A seção 4 apresenta os principais problemas observados pelo estagiário em cada etapa e sugestões de melhorias para cada uma delas.

A seção 5 detalha a relação entre as disciplinas cursadas pelo estagiário e as atividades realizadas durante os projetos.

A seção 6 apresenta a conclusão sobre o estágio desenvolvido.

2. ATIVIDADES DESENVOLVIDAS

Este capítulo tem por objetivo detalhar as atividades desenvolvidas durante o estágio supervisionado. Inicialmente são apresentadas as atividades de manutenção de sistemas e em seguida as de desenvolvimento.

2.1. MANUTENÇÃO DE SISTEMAS EM PHP


Diversas aplicações são desenvolvidas a fim de tornar atividades manuais mais simples, rápidas e automatizadas. Com intuito de que qualquer pessoa tenha acesso, sendo funcionário ou aluno, algumas aplicações possuem um subdomínio do domínio da faculdade e foram desenvolvidas em PHP, uma linguagem de programação web que atua no lado do servidor.

Dois projetos existentes foram alterados para aumentar a eficiência. O primeiro é o Caxxa, que não possui um link público, pois é de acesso exclusivo e pode ser acessado apenas pela intranet. O Caxxa permite o cadastro de novos eventos e cadastrar participantes para os mesmos.

O segundo projeto é o Acesso, que possui link público e pode ser acessado em <https://acesso.direitosbc.br/>. O Acesso consiste na recuperação de senha de um aluno para seu login nas aplicações da faculdade.

Ambos os projetos foram desenvolvidos em PHP e banco de dados MySQL. Para as manutenções necessárias, foi utilizado o software Notepad++.

A primeira necessidade no projeto Caxxa foi a ordenação das palestras por ordem crescente de data, sendo que a ordenação era feita por data de criação. Para isso foi acessado o código fonte e alterada a query de MySQL que ordenava por ID para data, retornando o resultado correto (*Figura 2.1*).



CAXXA
 CENTRO ACADÊMICO XX DE AGOSTO - FDSBC

[Palestras >> Inscrições](#)

Data	Horário	Local	Palestra	Prazo	Vagas
07/10/2015	17:45 - 18:40	Anfiteatro	PROCEDIMENTOS ESPECIAIS NO NOVO CPC	06/10/2015	260
15/10/2015	17:45 - 18:40	Anfiteatro	MULHERES E O DIREITO: ANÁLISE A PARTIR DOS CRIMES CONTRA A	14/10/2015	257

Faculdade de Direito de São Bernardo do Campo
 Autarquia Municipal
 2012 – Todos os direitos reservados

Figura 2.1 - Ordenação das palestras por data.

A segunda necessidade do projeto Caxxa foi de não permitir a alteração dos dados após o a data do evento. Esse problema foi recorrente, pois usuários preferiam editar todos os dados e assim tornar o antigo evento em um novo evento, a realizar a burocracia de criar um novo evento. Sendo assim, a lógica criada foi de comparar a data atual com a data do evento, caso fosse maior, o botão de atualizar deve ser bloqueado e apresentar a mensagem para o usuário (Figura 2.2).



CAXXA
 CENTRO ACADÊMICO XX DE AGOSTO - FDSBC

Palestras
Usuários
Sair

[Palestras >> Editar](#)

Título: PROCEDIMENTOS ESPECIAIS NO NOVO CPC

Descrição: PROCEDIMENTOS ESPECIAIS NO NOVO CPC

Palestrantes:

Local: Anfiteatro ▼

Data: 07/10/2015

Início: 17 ▼ 45 ▼ Término: 18 ▼ 40 ▼

Status: Aprovada

Inscrições: Abertas ▼ Prazo: 06/10/2015 

 Atualizar


 usuário : eder.rangel

Faculdade de Direito de São Bernardo do Campo
 Autarquia Municipal
 2012 – Todos os direitos reservados

Figura 2.2 - Bloqueio de atualização de eventos após a sua data de acontecimento.

No projeto Acesso, as necessidades foram de tratamentos de informações. O usuário segue um passo-a-passo para poder redefinir sua senha. A tela inicial apresenta um campo para preencher a matrícula, como é mostrado na *Figura 2.3*.

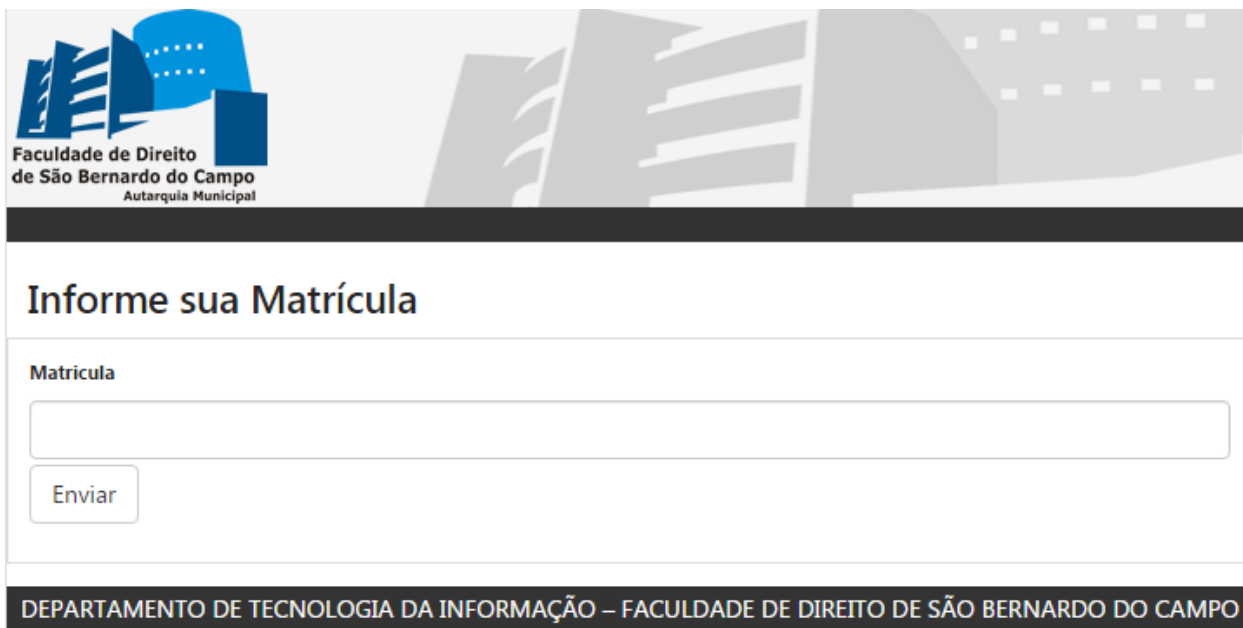
A imagem mostra a interface de login de um sistema web. No topo, há um cabeçalho com o logo da Faculdade de Direito de São Bernardo do Campo e uma ilustração de edifícios. Abaixo do cabeçalho, o título "Informe sua Matrícula" é exibido em uma fonte sans-serif. Segue um formulário com o rótulo "Matricula" e um campo de entrada de texto. Abaixo do campo, há um botão "Enviar". Na base da página, uma barra de rodapé contém o texto "DEPARTAMENTO DE TECNOLOGIA DA INFORMAÇÃO – FACULDADE DE DIREITO DE SÃO BERNARDO DO CAMPO".

Figura 2.3 - Tela inicial de Acesso com o campo de matrícula.

Nos passos seguintes, o usuário é orientado a confirmar seus dados para garantir a segurança e de que o próprio autor está tentando realizar a alteração. Os próximos passos são de confirmação de telefone e e-mail (*Figura 2.4*).

Com o Notepad++, foi procurado o código correspondente às validações dos campos de telefone e e-mail. A manutenção consistiu em facilitar a confirmação por parte do usuário. Em ambos os casos, foram desenvolvidas expressões regulares para permitir a entrada de dados e então foi excluída toda informação desnecessária para a verificação.



Digite Seu telefone Completo

Telefone

VoltarEnviar

Obs.: Caso o telefone informado esteja desatualizado ou incorreto, dirija-se ao Centro de Apoio ao Aluno para a atualização do cadastro.

98666XXXX

28666XXXX

DEPARTAMENTO DE TECNOLOGIA DA INFORMAÇÃO – FACULDADE DE DIREITO DE SÃO BERNARDO DO CAMPO

Figura 2.4 - Confirmação de um dos telefones cadastrados e e-mail em Acesso.

2.1.1. RESULTADOS

A visualização dos eventos criados no sistema Caxxa foi facilitada, uma vez que os usuários estão acostumados a pesquisar tuplas com data por esse atributo, do mais antigo para o mais recente, ou seja, os eventos que estão mais próximos da sua data de acontecimento aparecem primeiro na lista.

No sistema Acesso, anteriormente o usuário precisava digitar exatamente o mesmo número de telefone e e-mail cadastrados no sistema, mesmo se houvesse código local, hífen, etc. Após a modificação, a validação dos dados é de maior responsabilidade do sistema e não mais do usuário.

2.1.2. CONCLUSÃO

A manutenção no Caxxa inibiu usuários a burlarem a burocracia de criar um novo evento desde o início e da maneira correta. Sendo assim, todos os novos eventos precisam de autorização e acompanhamento para serem criados, divulgados e cadastrados seus respectivos participantes.

A manutenção do Acesso permitiu que usuários digitassem o telefone sem caracteres especiais ou código local e também digitassem apenas o usuário do e-mail, ao invés do e-mail completo, uma vez que todos possuem o e-mail institucional. As alterações reduziram o suporte para essa aplicação, pois os usuários possuem maior facilidade e menor chance de erros.

2.2. DESENVOLVIMENTO DO SISTEMA DE BOLSA DE ESTUDOS PROMAIS

Existem vários tipos de bolsas de estudos oferecidas pela Faculdade de Direito de São Bernardo do Campo: Para a Bolsa Auxílio, os estagiários bolsistas são admitidos mediante concurso público e recebem a Bolsa Auxílio que pode corresponder, no máximo, ao valor total da parcela da anuidade [3].

A Bolsa Mérito, premia com 11 parcelas da anuidade o calouro que tirou a nota mais alta no vestibular do período matutino e do período noturno e o aluno que obtiver a melhor média geral final em cada uma das turmas de cada série. A Bolsa Mérito propicia a isenção do pagamento da anuidade da série seguinte à do exercício de apuração, executando-se a 1ª parcela da anuidade e taxas de dependência, adaptação e pendências curriculares. Não concorrerão à Bolsa Mérito os alunos que obtiveram, na série de apuração, aproveitamento de estudos em virtude de disciplinas anteriormente cumpridas, assim como os que já usufruam de benefício similar outorgado por outra instituição pública ou privada [3].

Mediante convênio, o Santander S/A, disponibiliza acesso ao Programa Santander Universidades de Bolsas de Educação que tem por objeto a concessão de bolsas de educação a alunos que atendam os requisitos constantes dos Princípios Gerais do Programa Santander de Bolsas de Educação e que estejam matriculados em Instituições de Ensino Superior – IES que sejam parceiras do SANTANDER [3].

O desenvolvimento foi feito utilizando o sistema operacional Windows 7, software Microsoft Visual Studio 2013 para criação dos códigos-fonte em C# para programação do lado do servidor. HTML5, Javascript, JQuery, CSS3, Bootstrap e AngularJS [4] para a programação do lado do cliente. SQL Server para a criação do banco de dados.

O grande diferencial do projeto foi a utilização do AngularJS [4] que é um framework javascript open-source desenvolvido pelo Google. Possui padrão MVC e trabalha em conjunto com o HTML, tornando simples o desenvolvimento e uma aplicação muito dinâmica, tanto visualmente como no retorno de dados que são em JSON [5].

Todo o sistema foi desenvolvido utilizando o conjunto de ferramentas Bootstrap [6], que são, principalmente, de classes CSS e Javascript e facilitam na programação de layouts responsivos, priorizando que o usuário final possua uma ótima experiência ao utilizar as funcionalidades em qualquer dispositivo.

2.2.1. RESULTADOS

O banco de dados criado para a aplicação apresenta o digrama de relacionamentos (Figura 2.6). É possível perceber que as tabelas “Gabarito” e “Pessoa” são as principais devido a maior quantidade de relacionamentos.

O gabarito apresentado ao usuário é totalmente dinâmico, tanto no front-end como na atualização das informações. Um novo edital é lançado anualmente, sendo assim, novos requisitos são formados e novas questões são selecionadas ao usuário para que a aplicação apenas exiba as questões referentes ao ano atual.

As pessoas que são registradas na aplicação possuem várias informações por não serem apenas alunos da instituição, mas também familiares ou contribuintes à renda do aluno.

As tabelas que iniciam com “AspNet” são utilizadas para os usuários que utilizam a ferramenta com login e senha. Essas tabelas permitem manipular as permissões de acesso e controle das atividades de cada um.

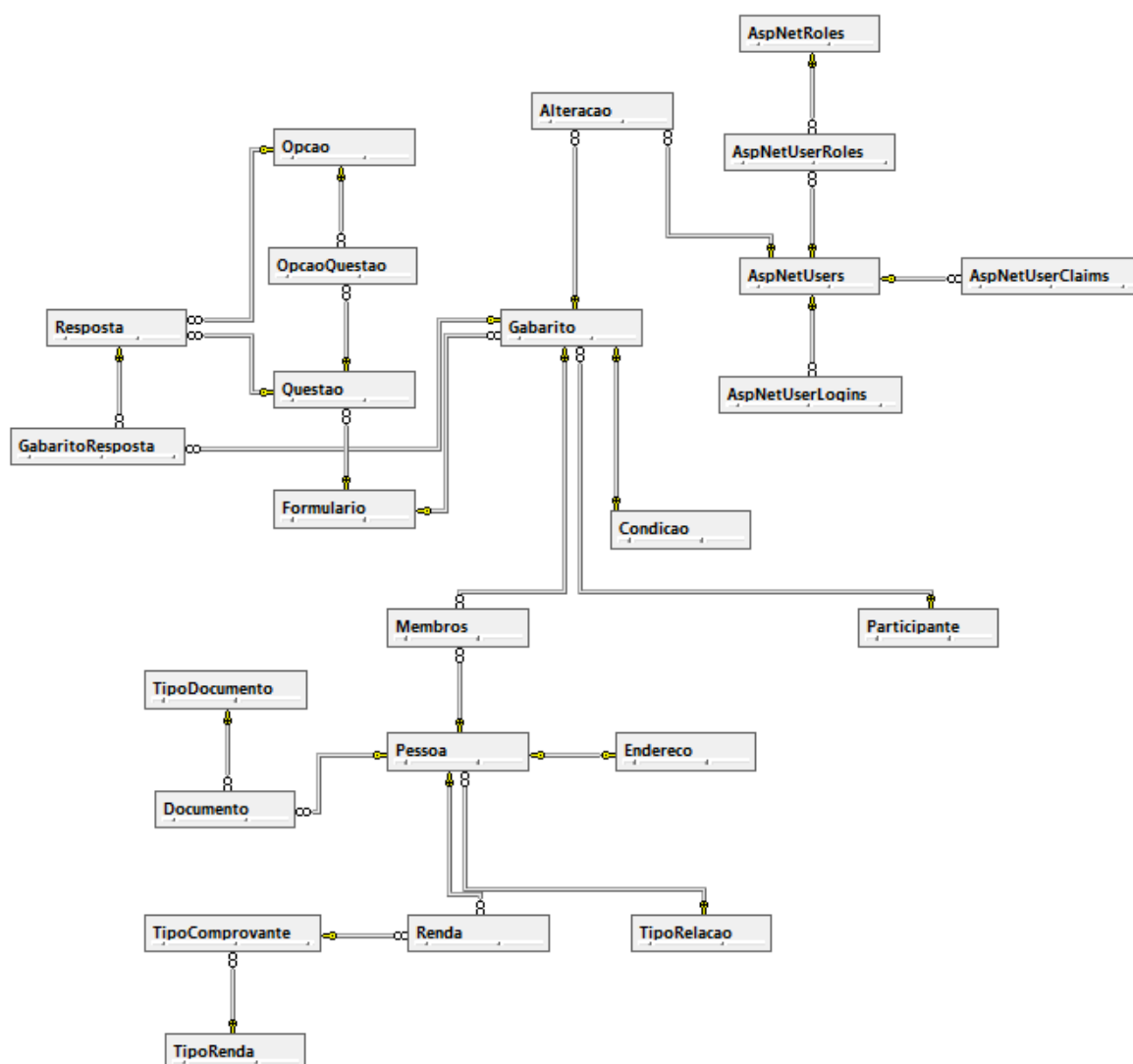
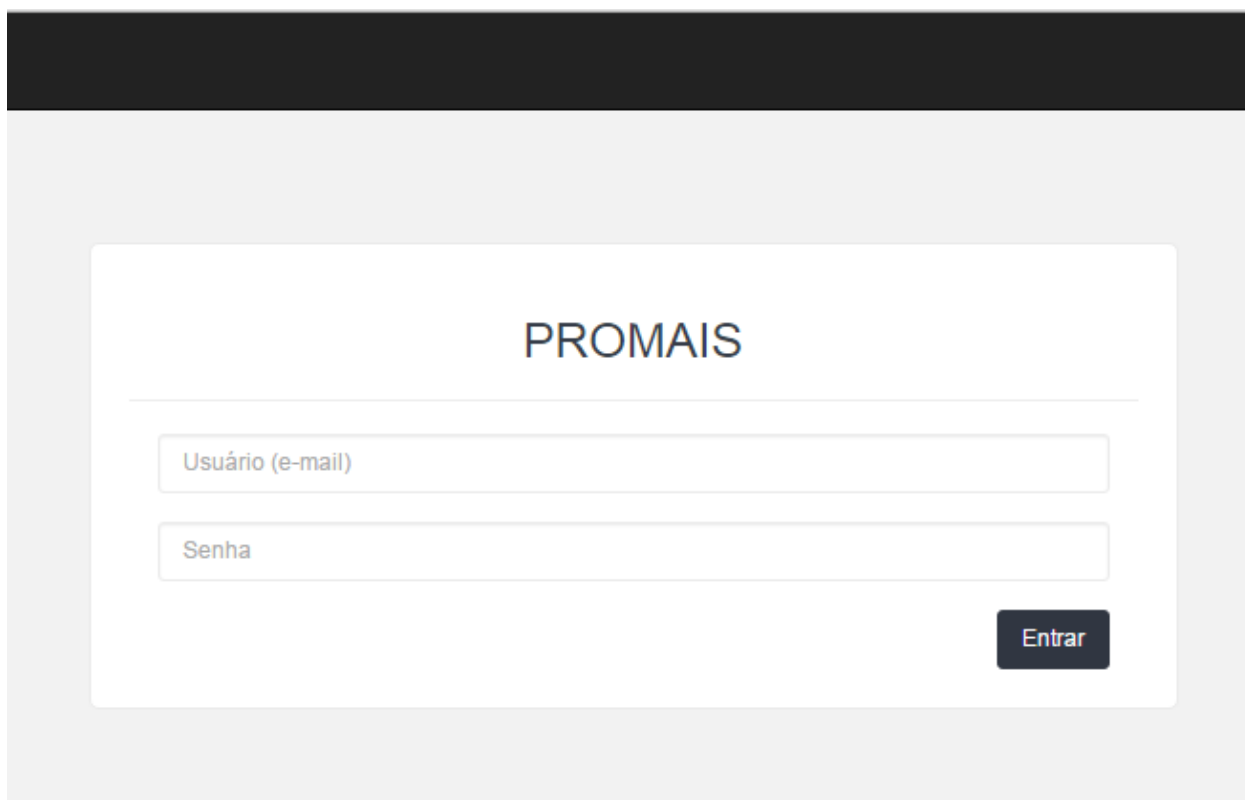


Figura 2.6 - Diagrama de relacionamento do banco de dados do PROMAIS.

A página inicial da aplicação exibe um formulário de login. Não é possível cadastrar pois o setor de Tecnologia da Informação é responsável pelo cadastro de pessoas autorizadas com a devida permissão de funcionalidades (Figura 2.7).

Após o login, a página exibe uma lista com todos os alunos que tiveram formulários cadastrados anteriormente. A lista exibe o número do processo, a matrícula, o nome, o status e a classificação. Há uma barra de pesquisa no topo que filtra por todos os campos simultaneamente. No canto inferior direito há um botão com opções para o usuário criar um novo formulário ou alterar sua senha (Figura 2.8).



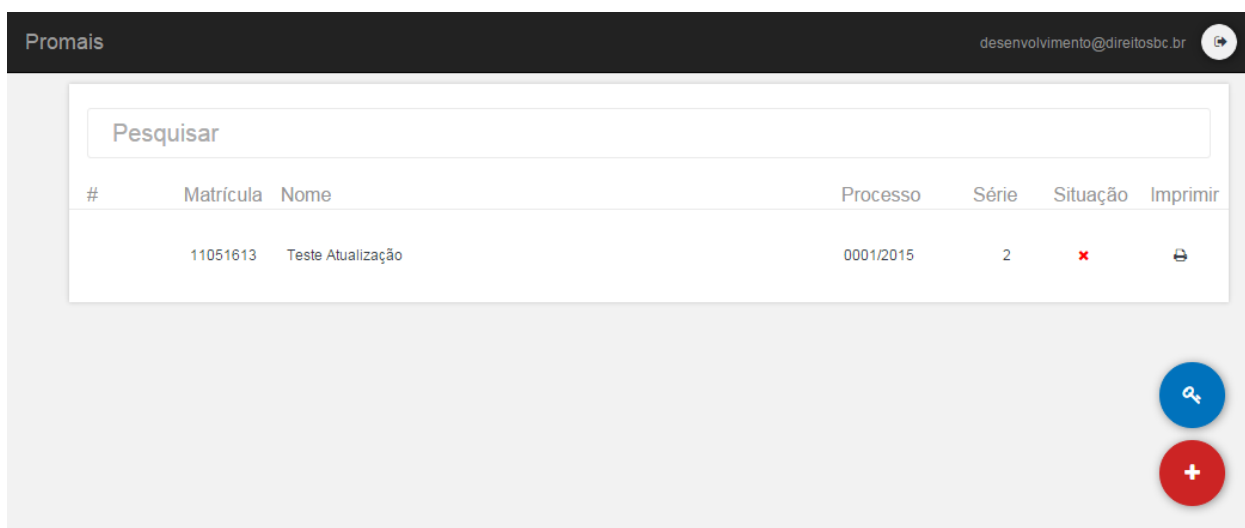
PROMAIS

Usuário (e-mail)

Senha

Entrar


Figura 2.7 - Tela de login da aplicação PROMAIS.



Promais

desenvolvimento@direitosbc.br

Pesquisar

#	Matrícula	Nome	Processo	Série	Situação	Imprimir
	11051613	Teste Atualização	0001/2015	2	×	






Figura 2.8 - Tela inicial da aplicação PROMAIS.

Ao selecionar um registro da lista ou para criar um novo registro, é exibida uma tela com um formulário detalhado ao lado esquerdo e um menu de navegação ao lado direito. No canto inferior direito, há opção de voltar para a lista de registros realizados, cancelar as alterações realizadas no formulário atual e a opção de salvar (*Figura 2.9*).

Com a utilização do AngularJS [4], o formulário se tornou totalmente dinâmico ao inserir, editar e remover pessoas, documentos, rendas, etc. Todas as informações são transformadas em JSON, ou seja, a maior preocupação é a estruturação dos dados para que seja recebido pela aplicação no back-end (C#) e gerenciada da maneira devida no banco de dados.

Promais desenvolvimento@direitosbc.br

1. Qualificação ⓘ

Processo* 0001/2015 Matrícula* 11051613 Série* 2

Nome* Teste Atualização

E-mail* teste.atualizado@direitosbc.br Telefone* (11) 3927-0255

2. Condições de Participação ⓘ

☒ Aluno do curso de Graduação ☐ Requerer inscrição no prazo
☒ Matriculado ☐ Apresentou documentação
☐ Não recebeu bolsa ou foi aprovado no ano anterior ☒ Não apresentou documentos inequivocamente falsos
☒ RFMF inferior ao limite (R\$ 1.277,00) ☐ Não omitiu informações

3. Outras Informações ⓘ

Tipo de Moradia **TM: 1** Membro familiar c/ doença grave **DG: 1**

☒ Própria ou cedida ☐ Não

1. Qualificação
 2. Condições de Participação
 3. Outras Informações
 4. Endereço Residencial
 5. Grupo Familiar
 6. Responsável Financeiro
 7. Comprovação de Rendimentos
 8. Observações

RFMPC: -1,000.0
RFMF: -1,000.0
Desclassificado ⓘ

Menu Edit Save

Figura 2.9 - Tela de alteração ou criação de registro.

Ao lado do título de cada segmento, existe um ícone de informações. Ao clicar nesse ícone é exibido em destaque as informações sobre aquele conteúdo, o mesmo contido no edital (*Figura 2.10*).

Outras Informações

4.2.3.5. PARA OBTENÇÃO DE PONTUAÇÃO POR CONDIÇÕES SOCIOECONÔMICAS DIFERENCIADAS OU PARA COMPROVAÇÃO DE PENSÃO ALIMENTÍCIA PAGA

a) Declaração pública ou particular, com firma reconhecida, ou outro comprovante idôneo, aceito pela CCB/FD mediante análise, comprovando moradia em habitação coletiva gratuita utilizada pelo grupo familiar, quando for o caso;

b) Contrato e último comprovante de pagamento de imóvel residencial financiado e utilizado como moradia do grupo familiar, quando for o caso;

c) Contrato e último comprovante de pagamento de imóvel residencial locado e utilizado como moradia do grupo familiar, quando for o caso;

d) Declaração pública ou particular, com firma reconhecida, ou outro comprovante idôneo, aceito pela CCB/FD mediante análise, comprovando moradia em habitação coletiva não gratuita utilizada como moradia pelo grupo familiar, quando for o caso;

e) Declaração pública ou particular, com firma reconhecida, ou outro comprovante idôneo, aceito pela CCB/FD mediante análise, comprovando moradia em área de ocupação irregular, área de manancial e outras moradias em situação de vulnerabilidade social utilizada pelo grupo familiar, quando for o caso;

f) Relatório médico recente (com data não anterior há 6 meses), indicando o C.I.D. (Código Internacional de Doença), caso um ou mais membros do grupo familiar, incluindo o aluno, seja portador de qualquer uma das doenças ou afecções relacionadas na Portaria Interministerial MPAS/MS nº 2.998, de 23 de agosto de 2001 (Anexo IV) deste Edital;

g) Declaração expedida por Instituição de Ensino Superior não gratuita que comprove matrícula ativa de um ou mais membros do grupo familiar na condição de pagante, exceto o aluno requerente; h) Histórico escolar ou outro documento que comprove que o aluno cursou, no mínimo, duas séries do ensino médio em escola pública. Esta pontuação somente será acrescida para alunos matriculados no primeiro ano do curso de graduação.

Fechar

Figura 2.10 - Informações do edital em destaque.

Ao clicar em salvar, o formulário é exibido em uma tela de resumo que é exibida como uma prévia de impressão. Os dados foram tratados e ajustados a uma folha A4. No canto inferior direito há um botão para o usuário voltar para a lista com todos os registros realizados, outro para editar o formulário atual e outro para imprimir (Figura 2.11).

Promais

desenvolvimento@direitosbc.br

Relatório Promais

Processo: 0001/2015 Matrícula: 11051613 Série: 2

Dados Pessoais

Nome: Teste Atualização Data de Nascimento: 14/07/2015
E-mail: teste.atualizado@direitosbc.br Telefone: (11) 3927-0255
RG: 16165165165

Condições de Participação

☒ Aluno do curso de Graduação ☒ Matriculado
☐ Não recebeu bolsa ou foi aprovado no ano anterior ☒ RFMF inferior ao limite (R\$ 1.277,00)
☐ Requeceu inscrição no prazo ☐ Apresentou documentação
☒ Não apresentou documentos inequivocamente falsos ☐ Não omitiu informações

Outras Informações

Tipo de Moradia: Própria ou cedida Membro familiar c/ doença grave: Não
Membro familiar c/ ens. sup. não gratuito: Não Egresso de escola pública: Não

Endereço Residencial

CEP: 09861-160 UF: SP
Logradouro: Avenida Molino Fabrini Município: São Bernardo do Campo
Número: 383 Comp.: Bairro: Independência

Grupo Familiar

Responsável Financeiro

Nome: Usuário Data de Nasc.: 13/09/1990 Parentesco: Irmão
CEP: 09861-160 UF: SP
Logradouro: Avenida Molino Fabrini Município: São Bernardo do Campo
Número: Comp.: Bairro: Independência

RFMPC: -1,000.0 RFMF: -1,000.0 Desclassificado

Voltar Editar Imprimir

Figura 2.11 - Tela de resumo e prévia de impressão.

2.2.2. CONCLUSÃO

O desenvolvimento do novo sistema de Bolsa de Estudos foi um teste para a introdução do estagiário ao AngularJS [4] e o produto final atingiu as expectativas. A estruturação do banco de dados foi desenvolvida de maneira correta e de acordo com as formas normais. A aplicação foi subdividida, tornando-a mais ampla que a anterior, porém muito mais compreensível ao usuário, pois há uma tela de listagem, outra de cadastro e outra de resumo. O AngularJS [4] proporcionou uma aplicação mais dinâmica e com menos códigos, pois o mesmo poderia ter sido feito com códigos em Javascript e DOM diretamente, mas o AngularJS [4], com uma programação correta, se encarrega de realizar essas operações sem repetições de códigos ou códigos extensos, reduzindo o tempo de produção e ampliando a produtividade dos usuários e desenvolvedores.

2.3. DESENVOLVIMENTO DO SISTEMA DE GESTÃO DE ATIVIDADES

Existem inúmeros métodos para organizar e gerenciar tarefas a serem realizadas. Pessoas e empresas procuram maneiras diferentes até se adequarem a uma específica, porém é algo muito subjetivo e pessoal. Métodos mais antigos como agenda, bloco de notas e post-it já foram e continuam sendo bastante utilizados, mas aos poucos têm perdido espaço para métodos digitais como aplicativos para smartphones e sites específicos para essa funcionalidade.

Um site bastante conhecido é o Runrun.it (*Figura 2.12*) [7]. Ele permite que uma equipe tenha bastante liberdade e recursos para gerenciar suas tarefas, assim como tempo de produtividade, designar a tarefa para membros específicos, indicativos gráficos de produtividade, relatórios e suporte em português.

O Runrun.it [7] e outros sistemas foram testados pela equipe, porém os recursos não se adequaram às necessidades específicas, portanto foi iniciado um projeto próprio com domínio público para gerenciar tarefas de qualquer lugar.

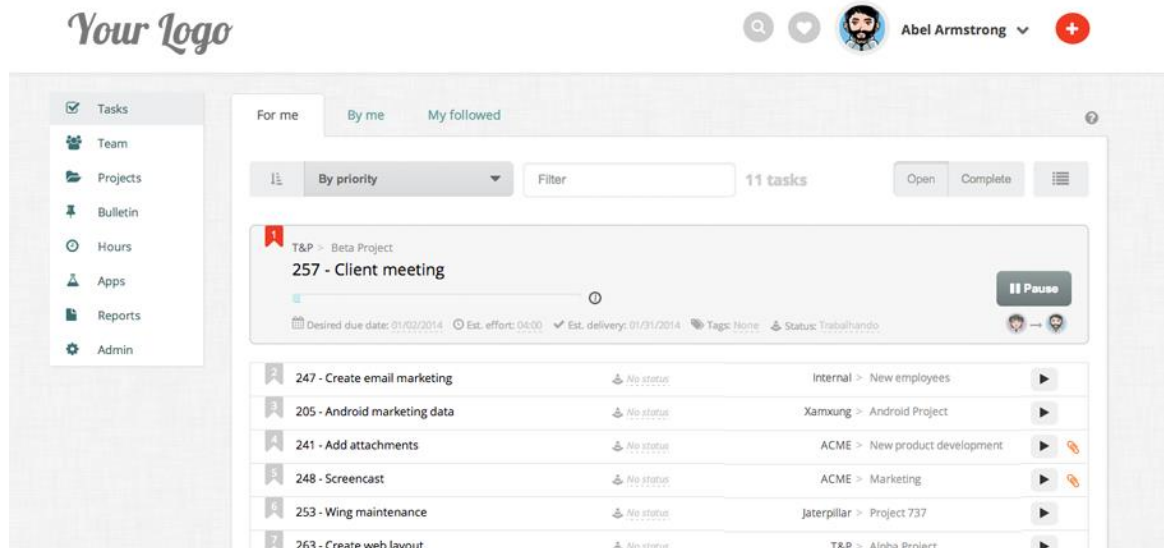


Figura 2.12 - Tela inicial do Runrun.it [<http://bit.ly/1OgVeKa>]

O aumento da produção e também popularização de dispositivos móveis, como smartphones e tablets, contribuíram muito para que houvesse melhorias nas ferramentas, tecnologias e técnicas para desenvolvimento de sites. Um exemplo que vem conquistando espaço são os sites chamados “Responsivos”. Diferente dos sites “Fluídos”, que apenas se adaptam ao tamanho do monitor em largura percentual, os sites “Responsivos” podem determinar quais conteúdos serão apresentados ao usuário e também a disposição dos objetos de acordo com a resolução do monitor que estiver utilizando, resultando em uma melhor navegação para o usuário em relação a ergonomia e tempo de processamento (Figura 2.13).



Figura 2.13 - Visualização de site responsivo em diversas resoluções de monitor [<http://bit.ly/1FgliiF>].

Inicialmente o projeto seria apenas de tarefas, mas foi expandido para a criação de tarefas e projetos. Todos tem uma base em comum, mas cada um tem sua particularidade. Tarefas possuem título, data de expiração, descrição, participantes, projeto e progresso. Projeto é um conjunto de tarefas.

O desenvolvimento foi feito utilizando o sistema operacional Windows 7, software Microsoft Visual Studio 2013 para criação dos códigos-fonte em C# para programação do lado do servidor. HTML5, Javascript, JQuery, CSS3 e Bootstrap para a programação do lado do cliente. SQL Server para a criação do banco de dados. Photoshop CS6 para a criação de banners e tratamento de imagens.

Primeiramente foi desenvolvida toda a lógica do sistema em uma base de dados sólida e eficiente. Algumas funcionalidades do back-end também foram desenvolvidas pelo estagiário, mas o principal foco foi a programação do front-end, ou seja, interligar os dados recebidos do lado do servidor (back-end), e apresentar todas as funcionalidades de forma amigável ao usuário final.

O projeto foi iniciado antes do projeto de Bolsa de Estudos PROMAIS sem a utilização de AngularJS [4] e com a utilização do Bootstrap [6]. Durante o desenvolvimento foi notada a necessidade de uma aplicação mais dinâmica e assim surgiu a ideia do AngularJS [4]. Como não havia experiência com a ferramenta, foi iniciado o projeto de Bolsa de Estudos PROMAIS como introdução e testes.

O projeto Controle de Atividades seria migrado para uma versão com AngularJS [4], mas muitos códigos seriam emendados e não seria uma boa prática, então foi iniciado um novo projeto para que tudo fosse desenvolvido no padrão correto. Ao iniciar o projeto, houve conhecimento do Angular Material Design [8], também desenvolvido pelo Google [9].

O Angular Material Design, segue os padrões [10] do sistema operacional Android Lollipop [11] encontrados em aplicativos do Google [9] como: Inbox [12], Music [13] e Keep [14]. Também é uma especificação para um sistema unificado de visual, movimento e interação que se adapta em diferentes dispositivos e diferentes tamanhos de tela, ou seja, substituiu o Bootstrap [6] na função de responsividade e design.

2.3.1. RESULTADOS

Uma demonstração do sistema foi gravada e disponibilizada no seguinte link: https://youtu.be/uXsNBhH_a40. Nela é possível ver as interações das funcionalidades com o usuário.

O banco de dados criado para a aplicação apresenta o seguinte digrama de relacionamentos (*Figura 2.14*):

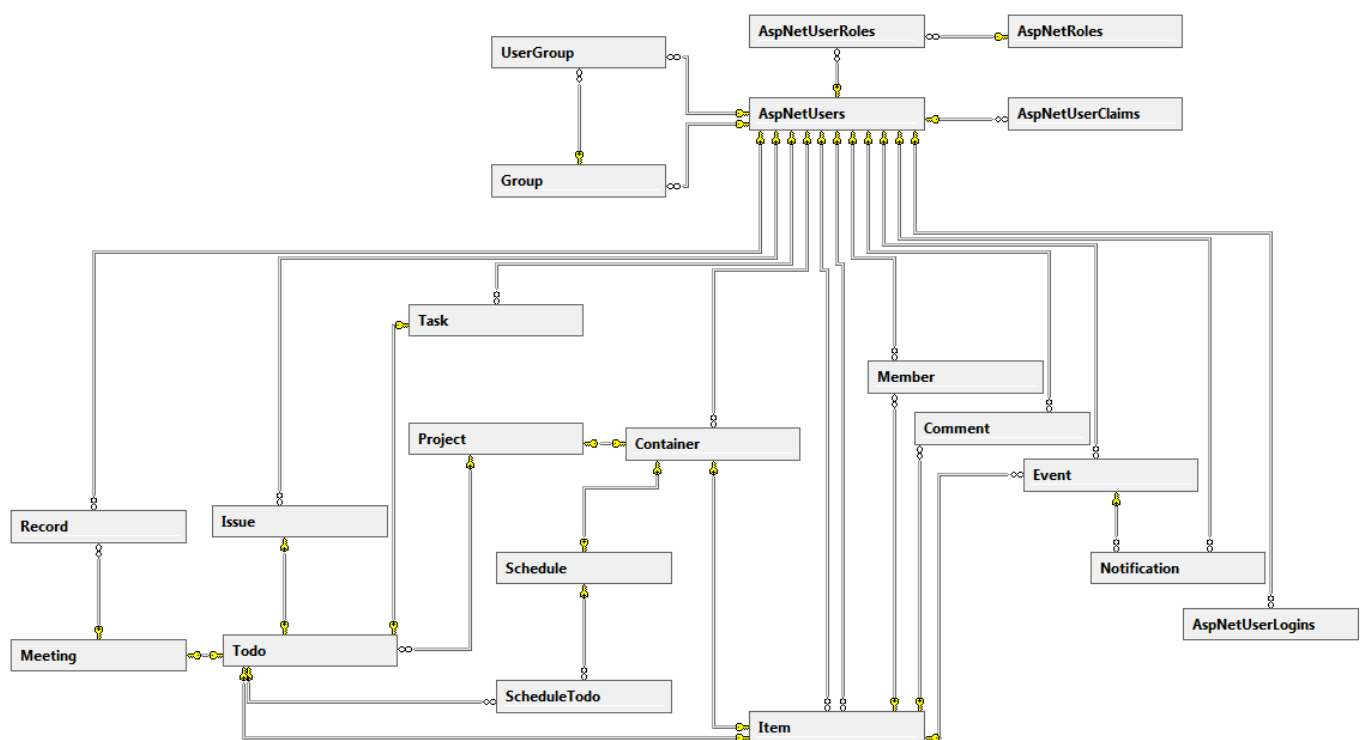


Figura 2.14 - Diagrama de relacionamento do banco de dados do Sistema de Gestão de Atividades.

A partir da *Figura 2.14* é possível perceber que as tabelas “AspNetUsers”, “Todo” e “Item” são as principais devido a maior quantidade de relacionamentos.

“AspNetUsers” são utilizadas para os usuários que utilizam a ferramenta com login e senha. Os usuários são totalmente ativos no sistema, pois são registrados ao criar uma atividade, editar, apagar, comentar e participar. As tabelas que possuem relacionamento permitem manipular as permissões de acesso e controle das atividades de cada um, além do sistema de

notificações por e-mail e no sistema e participação em atividades como espectadores, sem poder realizar alterações.

“Todo” é um objeto que herda suas propriedades para os objetos filho, sendo eles: tarefas, projetos, rascunhos, reuniões, planos de trabalho e chamados. Cada um possui uma particularidade que será explicada detalhadamente mais adiante.

"Item" representa cada objeto de atividade criada e se relaciona com os usuários e com as notificações.

A tela inicial da aplicação apresenta um formulário de login com algumas opções a mais, como: login com uma conta do Google [9], registro de nova conta manualmente pelo próprio usuário e envio por e-mail de uma nova senha em caso de esquecimento (*Figura 2.15*).

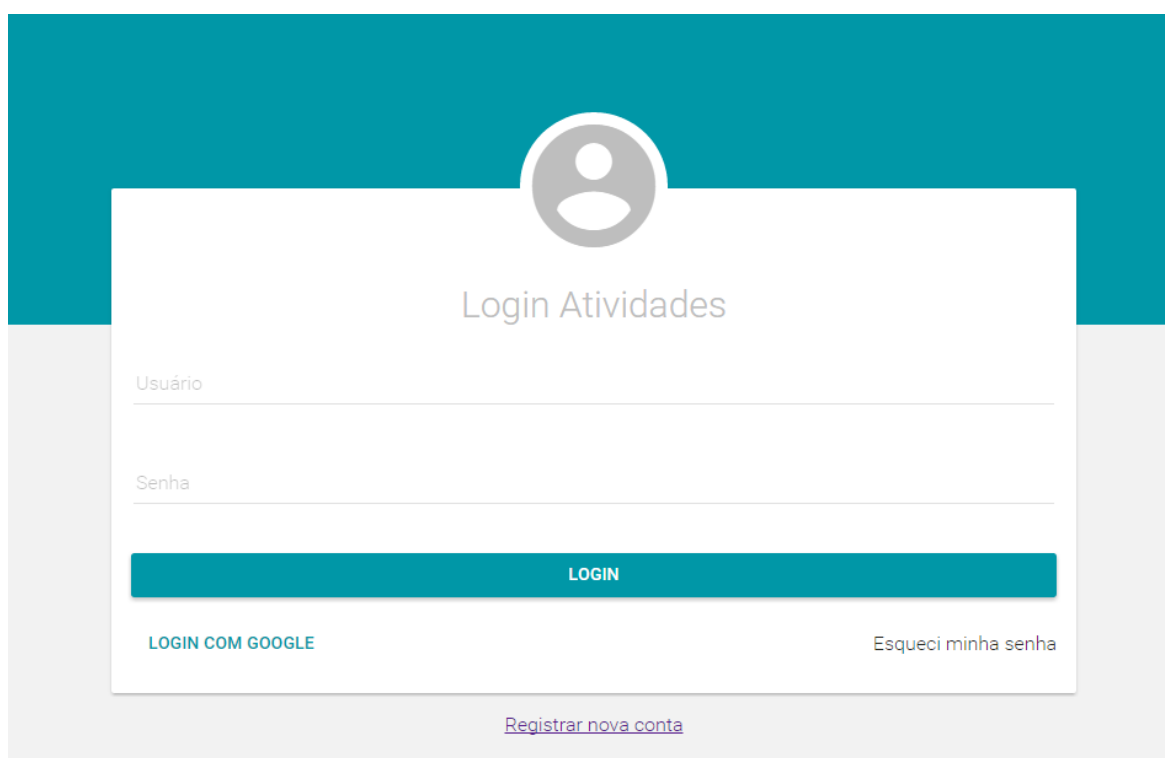
A imagem mostra a interface de login de um sistema. No topo, há uma barra decorativa de cor verde-azulada com um ícone circular cinza no centro. Abaixo, o título "Login Atividades" é exibido em uma fonte cinza. O formulário principal é branco e contém dois campos de entrada: "Usuário" e "Senha", ambos com sublinhados cinza. Abaixo dos campos, há um botão retangular de cor verde-azulada com o texto "LOGIN" em branco. Na base do formulário, há dois links: "LOGIN COM GOOGLE" em azul à esquerda e "Esqueci minha senha" em cinza à direita. Abaixo do formulário, no fundo cinza da página, há um link "Registrar nova conta" em roxo.

Figura 2.15 - Tela inicial do sistema.

Após efetuar o login, o usuário é direcionado ao sistema (Figura 2.16). A tela principal exibe um resumo de todas as atividades criadas e gráficos que são filtros para exibir na tela seguinte apenas o que foi especificado de acordo com o status. Os status possíveis são: Ativo, concluído, adiado e cancelado. Para uma melhor exibição, compreensão e tratamento de informações, foram criados alguns “pseudo-status”, ou seja, alguns status que não existem no banco de dados e não podem ser atribuídos diretamente a uma tarefa, mas são status criados no front-end baseados na data de expiração da tarefa. Sendo assim, o status “Ativo” foi subdividido em “Atrasado”, “Expirando” e “Em Andamento”. O status “Atrasado” é exibido quando a data expirou em relação à data atual. O status “Expirando” é exibido quando a data de expiração é menor ou igual a 3 dias em relação à data atual. O status “Em Andamento” é exibido quando a data de expiração é maior que 3 dias.

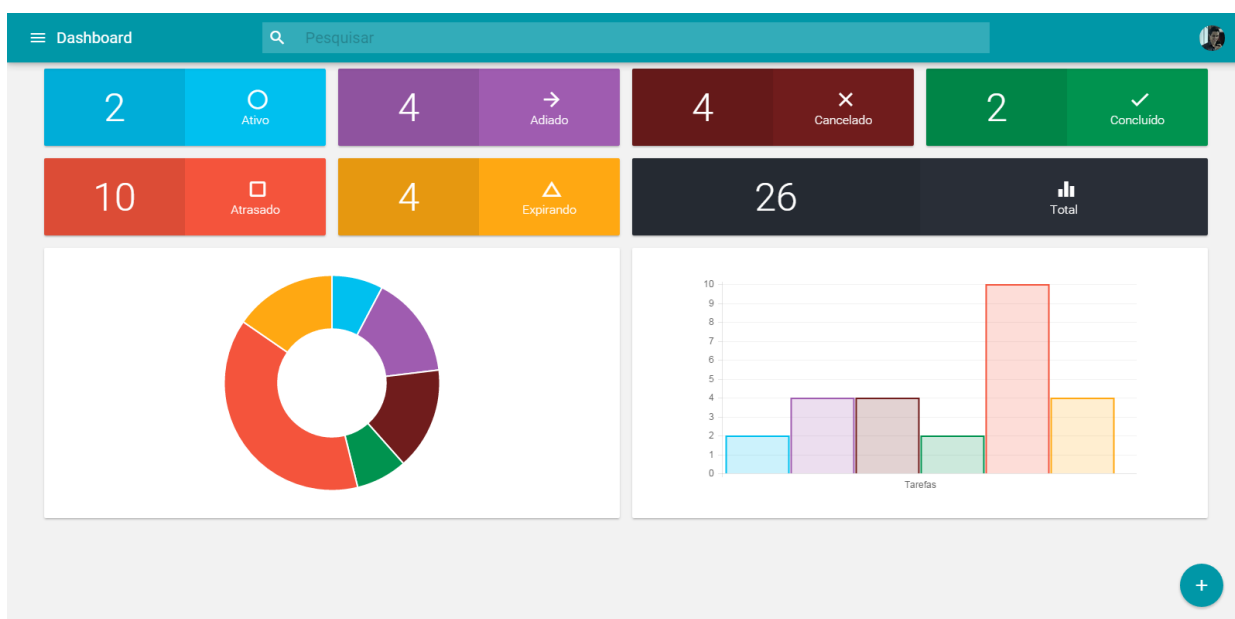


Figura 2.16 - Dashboard do sistema.

A tela de listagem de atividades exibe inicialmente as tarefas em ordem de data de expiração. Anteriormente, cada tarefa era exibida em uma linha e exibia um resumo das informações, sendo elas: ícone de status, código único da cada tarefa, ícone para alterar o status para “Concluído”, ícone para exibir os participantes ativos, ícone para excluir a tarefa, a data de expiração e o título.

Ao clicar em uma tarefa, o painel era expandido e exibia todas as informações com detalhes, como: menu em abas para ver os detalhes da descrição, comentários dos usuários sobre a tarefa e editar as informações existentes, rodapé com informação adicional de quem criou a tarefa (*Figura 2.17*).

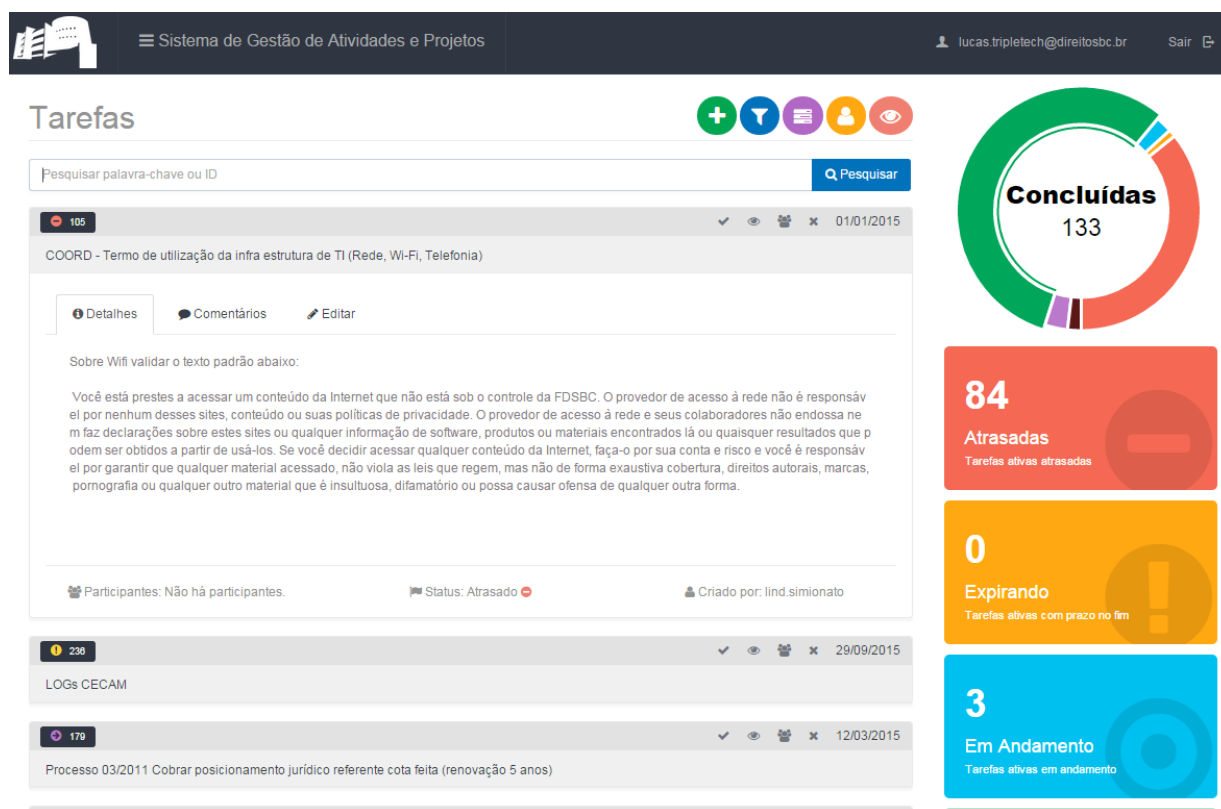


Figura 2.17 - Antiga tela de exibição dos detalhes, em abas, da tarefa quando expandida e filtros de status na barra lateral direita.

No desenvolvimento atual com Angular Material Design [8], a tela tem uma apresentação muito mais limpa, direta e com menos cores para trazer mais elegância e leveza à leitura do usuário. A antiga barra lateral direita com os status e servindo como filtro para o usuário foi retirada e inserida na Dashboard, como foi apresentado na *Figura 2.16*. A *Figura 2.18* apresenta a lista de atividades com a foto do criador, caso não possua, exibe a letra inicial do nome. O título da atividade, o nome de todos os participantes, uma barra de progresso, a data de expiração, o status e um botão de opções. O botão de opções permite excluir, colocar nos favoritos ou alterar como concluída.

O botão de criar novas atividades está localizado no canto inferior direito da tela, mantendo o padrão de aplicações com Angular Material Design [8]. O menu não é mais suspenso na barra superior, nela existe um campo de pesquisa, botão para filtrar, botão para alternar entre visualização em linha ou grade e a imagem de perfil do usuário ativo. O menu se tornou lateral e é exibido quando o usuário clica no botão específico, além de ser exibido em tablets de maneira similar à aplicativos (*Figura 2.18*).

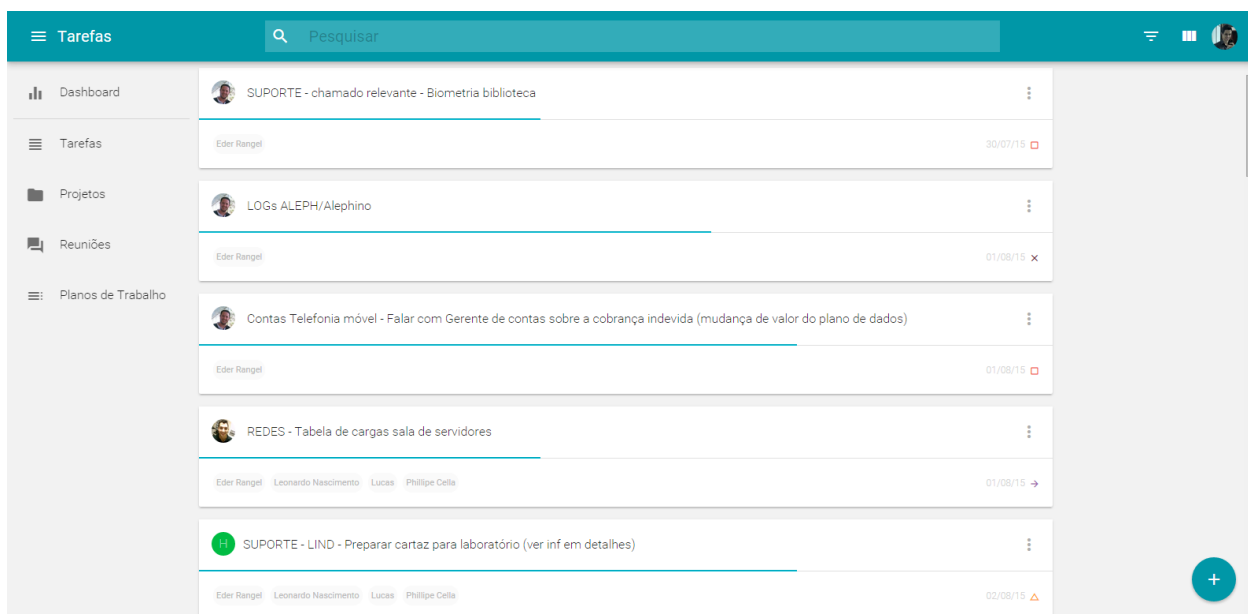


Figura 2.18 - Tela principal com as lista de atividades criadas em visualização de lista.

Anteriormente, os botões para adicionar nova atividade e filtrar se localizavam logo acima da lista de atividades. Ao clicar no botão de “nova tarefa”, abria um painel verde logo abaixo com todos os campos para especificar os detalhes da tarefa a ser criada (*Figura 2.19*).

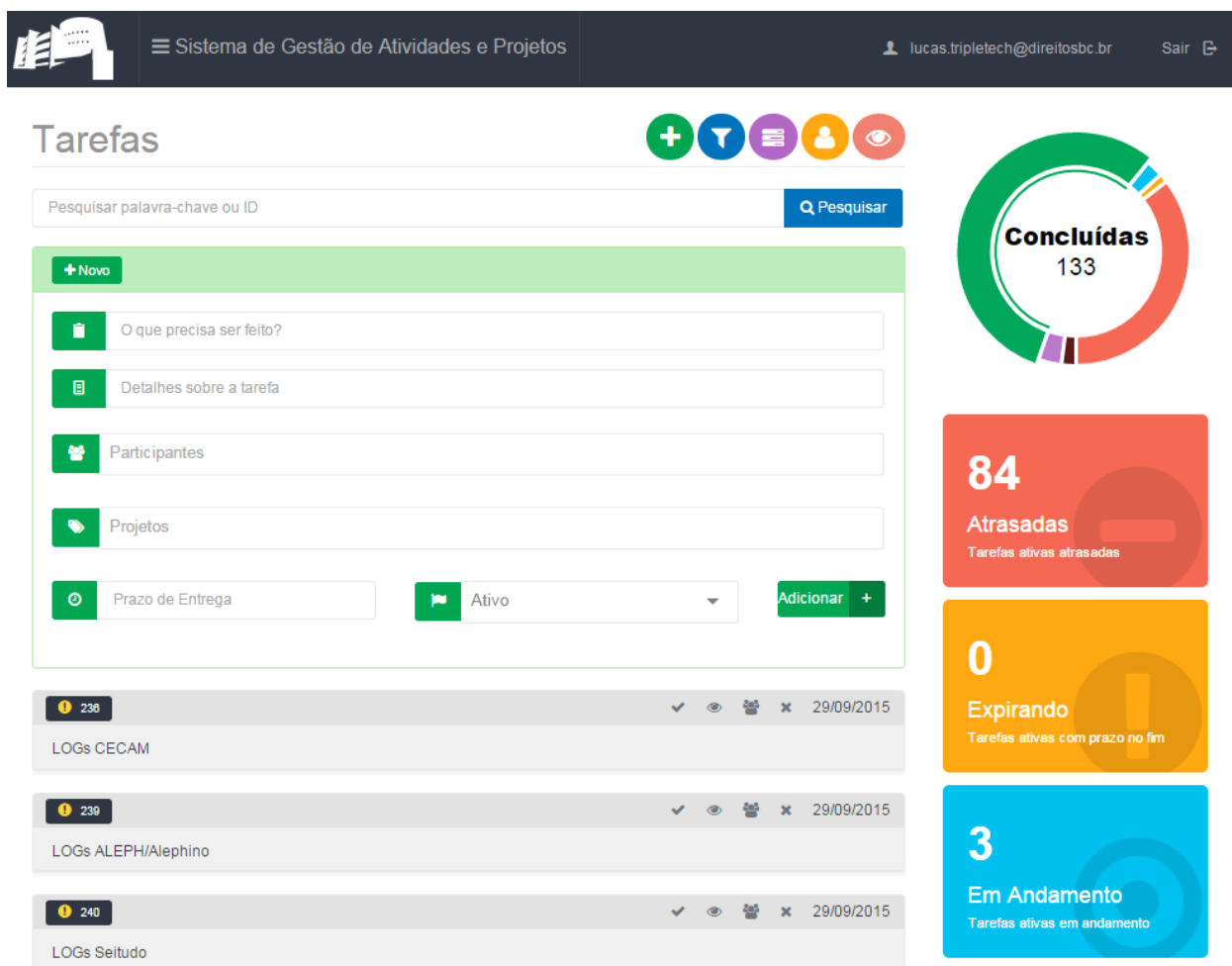


Figura 2.19 - Antiga tela principal com a opção de “nova tarefa” aberta e filtros de status na barra lateral direita.

No desenvolvimento atual com Angular Material Design [8], ao clicar no botão inserir, é apresentado um formulário flutuante em destaque. Ao clicar em alguma atividade, a mesma é apresentada em destaque e todas as informações são apresentadas à esquerda e os comentários à direita, diferente da versão anterior em que o conteúdo era apresentado na mesma tela e em abas (Figura 2.20). Na parte superior existem os botões de voltar para a lista, editar a atividade e opções, que contém as mesmas opções apresentadas na lista.

Nos formulários, os participantes apresentam fotos de perfil no momento de seleção e os status apresentam seus respectivos ícones, facilitando a manipulação de informações do usuário. Os comentários também apresentam a foto de perfil e só pode ser excluído pelo responsável.

Foi desenvolvida a funcionalidade “Desfazer”, em que ao realizar uma ação de alterar ou remover, o usuário recebe uma notificação para desfazer em um toast no canto inferior esquerdo, no caso de smartphones e tablets é exibido na parte inferior completa. Essa funcionalidade foi desenvolvida para evitar que o usuário sempre receba a mensagem “Deseja realmente fazer isso?” e clicar no botão “Ok” ou “Cancelar”. O “Desfazer” deixa a aplicação mais dinâmica e fluída.

Na barra superior, ao clicar no botão de filtro, abre um painel flutuante em destaque com os campos existentes no objeto de atividades, ou seja, é possível filtrar por participantes, projetos, status e autor da tarefa. O campo de autor permite escolher apenas uma opção. Os campos participantes, projetos e status permitem escolher várias opções. No caso de participantes, o filtro é realizado com o operador lógico “E”, ou seja, o resultado deve apresentar uma tarefa em que todos os participantes selecionados participem dela. No caso de projetos e status, o filtro é realizado com o operador lógico “OU”, ou seja, o resultado deve apresentar uma tarefa em que pelo menos uma das opções selecionadas estejam nela.

Caso seja feito um filtro com mais de um campo, o operador lógico utilizado para esse relacionamento é “E”, pois o resultado retornado é mais específico e não retorna tarefas que não são do interesse do usuário.

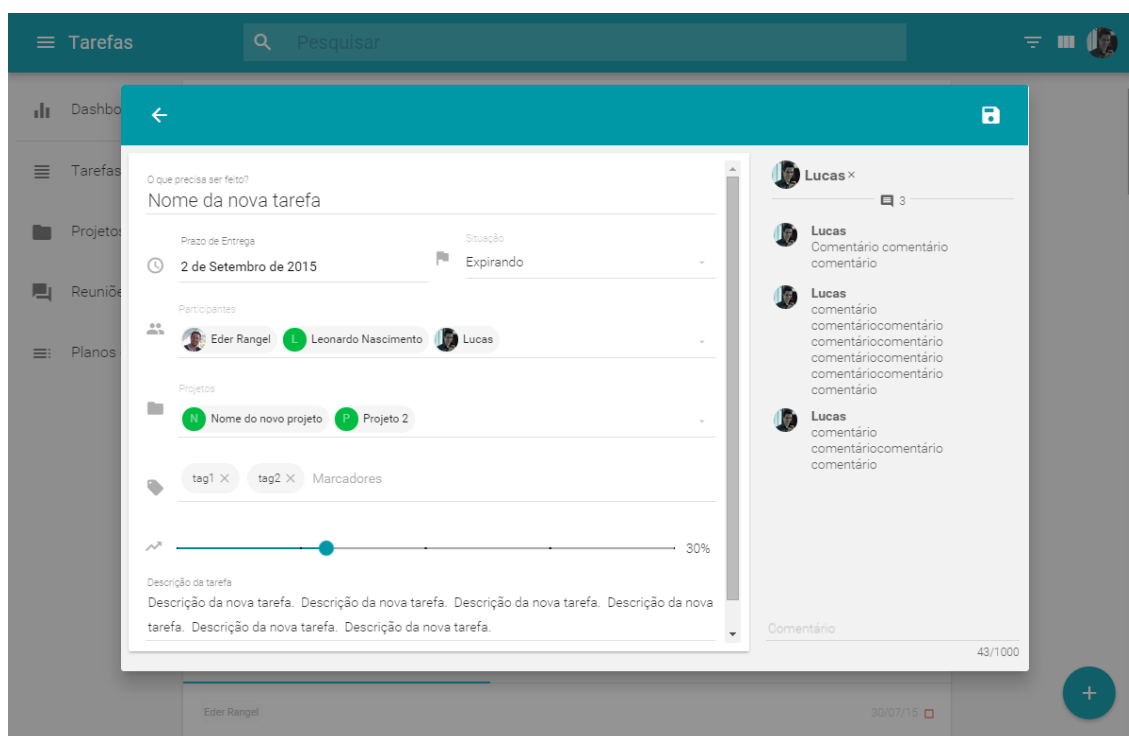


Figura 2.20 - Tela para edição de tarefa com visualização de comentários à direita.

Outro tipo de atividade que pode ser criada é "Projeto". Os projetos são um conjunto de tarefas, um agrupamento com um termo em comum. Uma tarefa pode possuir nenhum ou vários projetos. A criação do projeto é um pouco mais dinâmica que tarefa. Ao clicar no botão de criação, é aberta uma janela com o título do projeto, marcadores, descrição e abaixo uma linha apenas. A intenção dessa linha é que o usuário coloque todas as suas ideias e pressione "Enter" e, a cada vez pressionado, é criada uma nova linha abaixo para uma nova ideia. Após preencher todas as suas ideias, o usuário pode marcar no checkbox ao lado esquerdo da ideia para que ela se torne uma tarefa, caso não marque, essa ideia se tornará um Rascunho, uma categoria de atividade que possui apenas o título e o autor e pode ser transformada em uma nova tarefa futuramente. Ao lado esquerdo do checkbox há um ícone que permite a ordenação das ideias e ao lado direito há a opção de excluir a ideia e não criá-la em nenhuma categoria (*Figura 2.21*).

Ao clicar no botão salvar, um novo projeto é criado com o nome do usuário e todas as atividades marcadas são criadas como tarefas com a data atual, o projeto em questão, barra de progresso em zero por cento e status "Ativo". A barra de progresso do projeto será dinâmica, pois será calculada a média de progresso de todas as tarefas que pertencem a ele.

Nome do Projeto

Nome do novo projeto

marcador 1 X marcador 2 X marcador 3 X Marcadores

Descrição

Descrição sobre as atividades do novo projeto. 46/1000

O que precisa ser feito?

- ☒ Tarefa 1
- ☐ Rascunho 1
- ☒ Tarefa 3
- ☒ Tarefa 2
- ☐ Rascunho 2

Figura 2.21 - Tela para criação de projeto.

Foi analisado se as tarefas dos projetos seriam exibidas na mesma tela que a listagem de projetos, mas foi decidido que seriam exibidas em uma tela separada para melhorar a performance e para que haja uma nova URL, possibilitando que os usuários possam compartilhá-la. Portanto, há uma tela apenas para listagem de projetos que apresentam o nome e uma breve descrição e ao clicar no projeto, é aberta uma nova tela com o nome do projeto, marcadores e descrição e abaixo existem duas abas: Tarefas e Rascunhos. A aba de tarefas exibe as tarefas de maneira similar à listagem de tarefas individualmente e a aba de rascunhos exibe o autor, o título e um checkbox para selecionar e gerar uma nova tarefa a partir do rascunho selecionado (*Figura 2.22*)



Figura 2.22 - Visualização de rascunho de projetos.

O terceiro tipo de atividade que pode ser criada é "Reunião". Esse tipo de atividade foi criada para que houvesse um planejamento melhor antes de uma reunião e um controle melhor com a geração de atas após a reunião. A reunião possui um título, data, situação, participantes, marcadores, descrição e um campo, que pode ser formatado com um editor de texto avançado, para a ata (*Figura 2.23*).

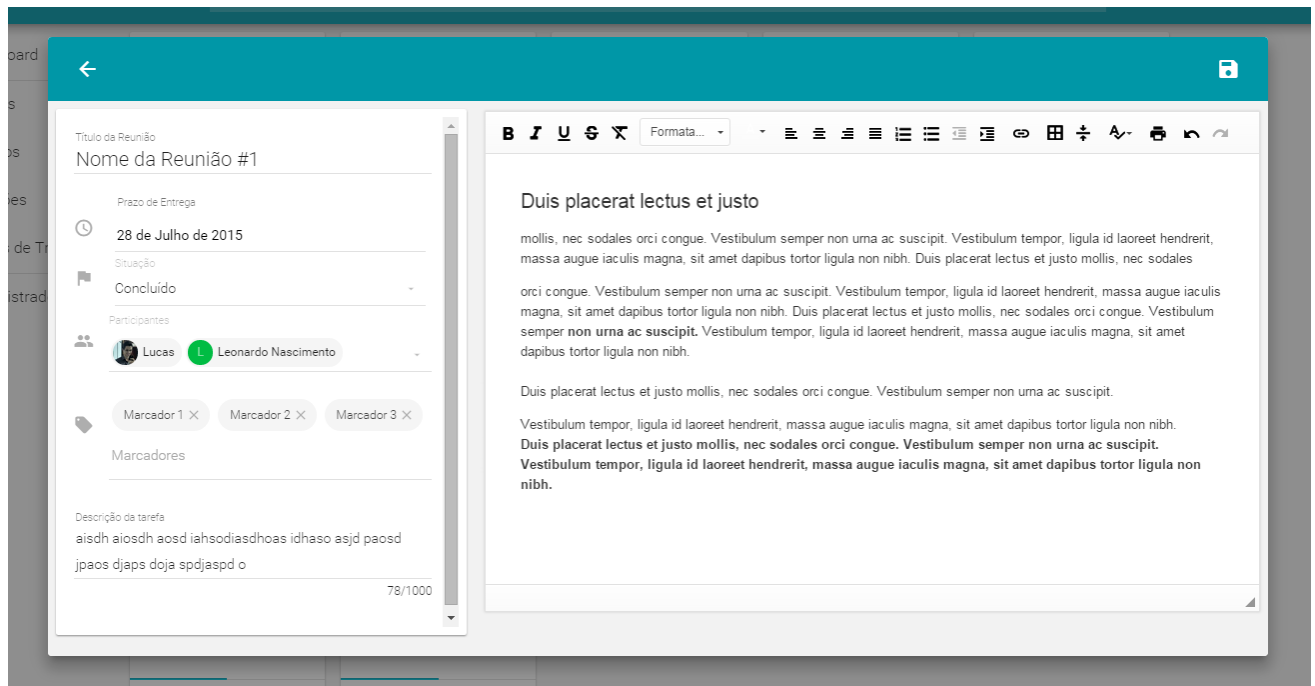


Figura 2.23 - Criação de Reunião.

O quarto tipo de atividade que pode ser criada é "Plano de Trabalho". Os planos de trabalho são parecidos com projetos, pois são conjunto de tarefas, um agrupamento com um termo em comum. Porém, ao criar um projeto novas tarefas e rascunhos são criados, enquanto ao criar um plano de trabalho são selecionadas tarefas já existentes, não possuem rascunho e o agrupamento por um termo em comum é realizado pela data de desenvolvimento. Os planos de trabalho se baseiam na ideia de Sprints, que são pequenos ciclos que formam a metodologia de desenvolvimento Scrum [15]. A criação de Plano de Trabalho é realizada com o título, descrição, datas de início e fim e a seleção de tarefas existentes (*Figura 2.24*). A listagem das atividades do plano de trabalho é semelhante ao de projetos, mas sem a aba de rascunhos.

The screenshot displays a mobile application interface for creating a new project plan. The top bar is teal with a hamburger menu icon, the text "Planos de Trabalho", a search bar with the placeholder "Pesquisar", and a user profile icon. Below this, a modal form is shown with a teal header containing a back arrow and a save icon. The form fields are as follows:

- Nome do Plano de Projeto:** "Criação do novo plano de projeto"
- Descrição:** "Descrição do novo plano de projeto" (with a character count of 34/1000)
- Data de Início:** "13 de Outubro de 2015" (with a clock icon)
- Data de Fim:** "31 de Outubro de 2015" (with a clock icon)
- Tarefas:** A list of tasks with user avatars and status indicators (green circles with 'S'). The tasks are:
 - BPM - Termo de referência...
 - CAXXA - bilhetagem copiada...
 - COORD - Termo de utilizaç...
 - Contas Telefonia móvel - ...

A teal circular button with a white plus sign is located at the bottom right of the screen.

Figura 2.24 - Visualização de criação de plano de trabalho.

Outro tipo de atividade que será criada futuramente é "Chamado". Os chamados serão utilizados para atividades não programadas, ou seja, são tarefas que podem ocorrer ocasionalmente, não pertencem a algum tipo de projeto e, geralmente, possuem uma data de conclusão curta.

2.3.2. CONCLUSÃO

O sistema está sendo desenvolvido com as ferramentas mais modernas do momento, possibilitando atingir os novos objetivos e um layout leve e agradável aos olhos do usuário final. O layout responsivo permite que tenham grande liberdade para poder acessar o sistema em desktops, tablets e até mesmo smartphones. O domínio público permite o usuário a gerenciar

suas tarefas de qualquer lugar, facilitando o uso de usuários com horários flexíveis e diferenciados.

O código ficou mais elegante com o AngularJS [4], considerando que todas as novas informações eram atualizadas manualmente com jQuery à cada ação, a praticidade de tratar os itens como objetos torna o AngularJS [4] indispensável para a manipulação das informações.

O sistema está rápido, mas deve melhorar com a implementação de métodos em AJAX. O carregamento parcial de pequenos trechos de código pode fazer muita diferença no projeto como um todo.

A aceitação dos usuários de testes foi extremamente positiva, principalmente em comparação ao antigo projeto, considerando que pequenos erros são alertados e corrigidos antes que a próxima versão seja publicada no servidor.

O sistema ainda está em fase inicial, mas já possui mais de 130 tarefas criadas. Provando que seu desenvolvimento é de grande importância e prioridade para a empresa, uma vez que detalhes como, por exemplo, seus próprios campos e status personalizados, o diferem de sistemas prontos, similares ao Runrun.it.

2.4. DIAGNÓSTICO DOS PRINCIPAIS PROBLEMAS OBSERVADOS E SUGESTÕES DE MELHORIA

Os sistemas em PHP são de longa data e poderiam ser desenvolvidas novas versões com mais funcionalidades e layout responsivo. Existem famosos CMS em PHP que poderiam trazer benefícios aos sistemas, como o Wordpress, ou então a utilização de algum framework, como o Zend.

O sistema de Bolsa de Estudos PROMAIS era muito antigo e desenvolvido em Microsoft Access, ou seja, muito defasado e impróprio para a utilização em uma empresa. A solução foi o desenvolvimento de um novo sistema desde o início com o Microsoft Visual Studio, C#, AngularJS, SQL Server e orientação à objetos. Algumas implementações ainda podem ser realizadas como perfis de usuários e permissões, especificando que alguns usuários possam editar e outros apenas visualizar as informações.

O Sistema de Gestão de Atividade teve um grande avanço entre as duas primeiras versões. Novas funcionalidades foram inseridas devido às novas ferramentas e tem grande potencial. Melhorias podem ser inseridas, como: novos filtros, edição de comentários, gráficos dinâmicos, sistema de notificações locais e por e-mail, perfis de usuários com permissões por setor e individual.

3. FUNDAMENTAÇÃO TEÓRICA

As disciplinas cursadas contribuíram de várias formas, direta ou indiretamente, nas atividades realizadas no período de estágio. A Tabela 3.1 apresenta as disciplinas cursadas pelo aluno e que foram necessárias para o desenvolvimento das atividades.

Tabela 3.1 - Disciplinas desenvolvidas durante o estágio com os suas respectivas abordagens.

Algoritmos e Estrutura de Dados	<ul style="list-style-type: none"> • Ordenação.
Análise de Algoritmos	<ul style="list-style-type: none"> • Custo de operações.
Banco de Dados	<ul style="list-style-type: none"> • Banco de Dados Relacional; • Diagrama Entidade-Relacionamento; • Diagrama de classes; • Consultas SQL.
Engenharia de Software	<ul style="list-style-type: none"> • Planejamento; • Modelo Entidade-Relacionamento; • Padrão MVC.
Linguagens Formais e Autômata	<ul style="list-style-type: none"> • Expressão Regulares.
Lógica Básica	<ul style="list-style-type: none"> • Operadores Lógicos.
Processamento da Informação	<ul style="list-style-type: none"> • Lógica de programação.
Programação Orientada a Objetos	<ul style="list-style-type: none"> • Paradigma orientado a objetos.
Programação para Dispositivos Móveis	<ul style="list-style-type: none"> • HTML; • Responsividade.
Programação para Web	<ul style="list-style-type: none"> • HTML; • CSS; • Javascript; • Servlet API.
Segurança de Dados	<ul style="list-style-type: none"> • Criptografia; • Autenticação.

3.1. PLANEJAMENTO DE PROJETOS

Segundo os pontos citados na **Engenharia de Software** e nos livros de Somerville [15] e Pressman [16], não basta apenas programar e desenvolver projetos, é necessário ter um

planejamento completo antes de iniciar a programação em si, além de possuir desenvolvedores competentes na equipe. Todo sistema está sujeito a falhas, mas quando há bom planejamento, incluindo os imprevistos, dificilmente terá grandes problemas durante o período de utilização, nada que o comprometa seriamente. Tais fatos foram notados ao desenvolver os sistemas de Bolsa de Estudos PROMAIS e Sistema de Gestão de Atividades. O primeiro foi totalmente planejado, desde a estrutura das classes como quais ferramentas seriam utilizadas e o projeto foi desenvolvido do começo ao fim como o previsto. O segundo projeto havia sido planejado, mas durante o desenvolvimento foi notada muita repetição de códigos e a necessidade de uma ferramenta que tornasse a aplicação mais dinâmica e suprisse esse obstáculo, e assim foi pesquisado e encontrado o AngularJS [4].

3.2. DESENVOLVIMENTO WEB

3.2.1. HTML

Em **Programação para Web** foram desenvolvidas aplicações para web e em **Programação para Dispositivos Móveis** foram vistos conceitos semelhantes ao HTML. O HTML5 é uma linguagem que foi desenvolvida para exibir conteúdos em páginas web. A diferença para algumas outras linguagens de programação é que o HTML5 não necessita ser compilado, ele é apenas interpretado pelo navegador. A programação é realizada através de tags (<tag>) que podem possuir atributos e elementos filhos, desde que estejam dentro da tag de fechamento, que é representada por uma barra (</tag>). A *Figura 3.1* demonstra a estrutura de um arquivo HTML com algumas tags novas que foram introduzidas na versão 5, como header, nav, section, article, aside e footer. Novos elementos foram introduzidos para dar semântica à linguagem, ou seja, ao ler um código é possível ter noção do conteúdo que será apresentado de acordo com as tags utilizadas, nas versões anteriores, todo o conteúdo possuía o mesmo nível de informação.

Com base na *Figura 3.1*, é demonstrada a herança de tags. As tags que são apresentadas dentro de uma tag de abertura e fechamento, são consideradas filhas, como h1, form e nav que são filhas da tag header.

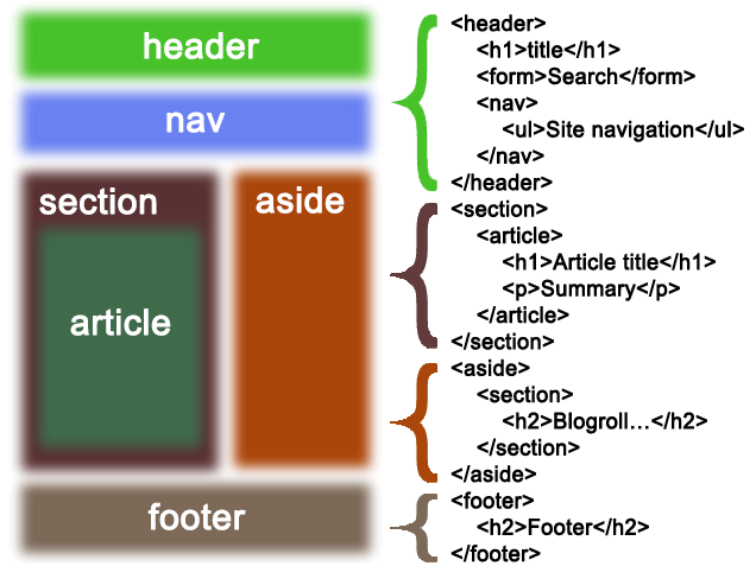


Figura 3.1 - Estrutura do HTML com novas tags introduzidas na versão 5.

O HTML5 é responsável pela exibição de conteúdo, sendo assim, não é capaz de criar novos elementos, conectar com um banco de dados ou com algum servidor, por exemplo. Essas outras atividades são realizadas por outras linguagens de programação que caminham em conjunto com o HTML5.

3.2.2. JAVASCRIPT

A linguagem de programação Javascript permite que haja interação entre o conteúdo HTML5 e o usuário, tornando a conteúdo da aplicação dinâmico. O DOM é uma multi-plataforma que representa como as marcações em HTML, XHTML e XML são organizadas e lidas pelo navegador. Uma vez indexadas, estas marcações se transformam em elementos de uma árvore que você pode manipular via API, ou seja, o Javascript é capaz de manipular essas informações. Após o HTML ser carregado e interpretado pelo navegador, ele mesmo não tem poder para alterá-lo, mas o Javascript pode inserir, alterar ou remover nós em tempo de execução (Figura 3.2) [17].

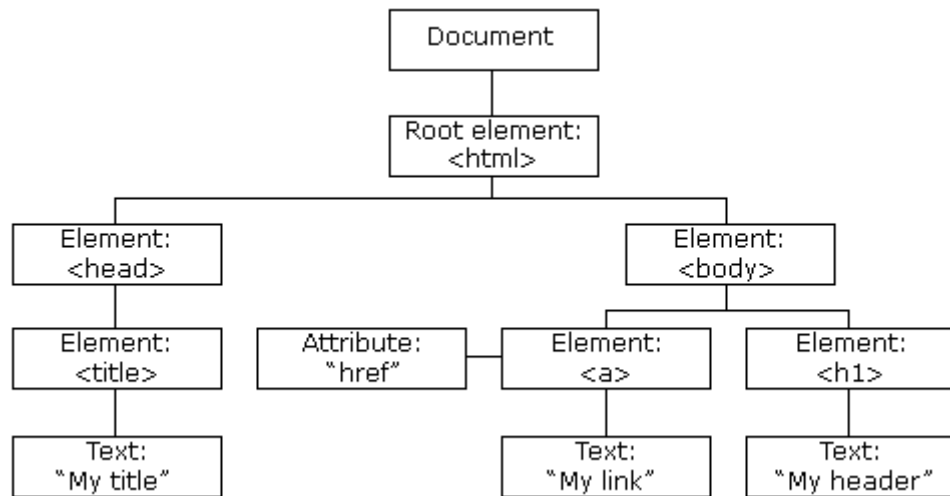


Figura 3.2 - Estrutura de uma árvore DOM

A *Figura 3.3* apresenta um exemplo de manipulação de uma árvore DOM com o Javascript. A primeira linha cria uma tag do tipo “div” e salva na variável “el”. A segunda linha acrescenta a variável “el” ao elemento “body”. A terceira linha acrescenta uma classe CSS “container” ao elemento “el”. A quarta linha adiciona uma margem superior de 30px ao elemento.

Se esse código fosse desenvolvido diretamente pelo HTML5, seria o mesmo que:
`<body><div class="container" style="margin-top: 30px"></div></body>`

```

var el = document.createElement("div");
document.body.appendChild(el);
el.className = "container";
el.style.marginTop = "30px";
  
```

Figura 3.3 - Exemplo de manipulação de DOM com Javascript.

3.2.3. ANGULARJS

O AngularJS [4] foi o grande encarregado por essa função de manipular o DOM nos sistemas de Bolsa de Estudos PROMAIS e o de Gestão de Atividades. A única necessidade do desenvolvedor é se preocupar com o JSON gerado e o AngularJS [4] o interpreta e cria o DOM de acordo com as informações que recebe.

A Figura 3.4 é uma implementação básica em AngularJS [4] para a criação de um novo nó com o conteúdo desejado. O primeiro bloco é o código HTML em que a tag “li” possui uma instrução “ng-repeat” do AngularJS [4] para adicionar o conteúdo “text” da lista “opts” enquanto houver filhos. O código abaixo é o Javascript responsável por alimentar o JSON e adicionar o valor “x” ao clicar no botão “Add”. À direita está o resultado final da aplicação de exemplo.

Esse mesmo código em Javascript puro seria um pouco mais complexo, mas nesse caso foi necessário acrescentar um novo valor no JSON e a aplicação se encarrega de acrescentar um novo nó automaticamente.



Figura 3.4 - Exemplo de criação de nó com AngularJS.

3.2.4. CSS

O HTML5 possui atributos para personalizar as tags adicionadas, porém é um método depreciado e pouco eficiente com o surgimento do CSS3.

O CSS3 é uma linguagem de folha de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como HTML5 e tem como papel principal separar o conteúdo de um documento e o seu formato.

A *Figura 3.5* apresenta duas maneiras de como inserir personalizações CSS a um elemento: por classe e inline. A classe pode ser atribuída a vários elementos e assim todos terão apresentação em comum. Ao alterar uma classe, todos os elementos serão afetados e terão sua apresentação modificada. Os atributos da classe podem ser escritos no mesmo arquivo ou em um arquivo externo e importado. O método inline é adicionado com o “style”, é atribuído apenas ao elemento em questão e, caso haja alguma classe no elemento, sobrepõe os atributos coincidentes da classe.



Figura 3.5 -Métodos de utilização do CSS [http://bit.ly/1G00zSp].

3.3. LÓGICA DE PROGRAMAÇÃO

3.3.1. CONEXÃO COM BANCO DE DADOS

Todo o conteúdo até o momento está do lado do cliente em JSON, não foi enviado nada para o servidor e não foi feita nenhuma persistência de informações, portanto as informações são

perdidas ao fechar a aplicação. Para a conexão com o banco de dados foi utilizada a linguagem C# da Microsoft, ela é uma linguagem orientada à objetos, fortemente tipada, foi baseada no C++ e possui similaridades com Java, linguagem desenvolvida em **Processamento da Informação, Programação Orientada a Objetos** e conceitos vistos em **Lógica Básica, Algoritmos e Estrutura de Dados e Análise de Algoritmos**.

A *Figura 3.6* é um exemplo de conexão básica com o banco de dados na linguagem C# em que a conexão é declarada, em seguida são adicionados os parâmetros da conexão, a conexão é aberta, uma mensagem é exibida, a conexão é fechada e outra mensagem é exibida.

```
System.Data.SqlClient.SqlConnection con;

private void Form1_Load(object sender, EventArgs e)
{
    con = new System.Data.SqlClient.SqlConnection();

    con.ConnectionString = "Data Source=.\SQLEXPRESS;
AttachDbFilename=C:\MyWorkers.mdf;Integrated Security=True;Connect
Timeout=30;User Instance=True";

    con.Open();

    MessageBox.Show("Database Open");

    con.Close();

    MessageBox.Show("Database Closed");
}
```

Figura 3.6 - Exemplo de conexão básica com o bando de dados em C#.

3.3.2. EXPRESSÕES REGULARES

Foram utilizadas expressões regulares, conteúdo de **Linguagens Formais e Autômatas**, no Sistema de Bolsa de Estudos PROMAIS para verificar se os valores inseridos pelos usuários estavam de acordo com as informações necessárias, por exemplo: número de telefone não pode aceitar caracteres ou caracteres especiais, mas pode aceitar hífen, parênteses e espaço. Sendo assim, a expressão regular que representa uma maneira correta para telefone é exibida na *Figura 3.7*.

Não há uma única expressão regular para alguns casos. Esse é um caso que depende da forma que o desenvolvedor decide para um melhor desempenho na sua aplicação. O exemplo apresentado permite receber telefones, obrigatoriamente, com DDD, mas sem obrigação dos parênteses. É permitido colocar espaço entre o DDD e o número ou preenchê-lo completamente sem espaço. No meio do telefone é aceito o sinal de hífen, ponto ou espaço. São aceitos telefones com oito ou nove números no total.

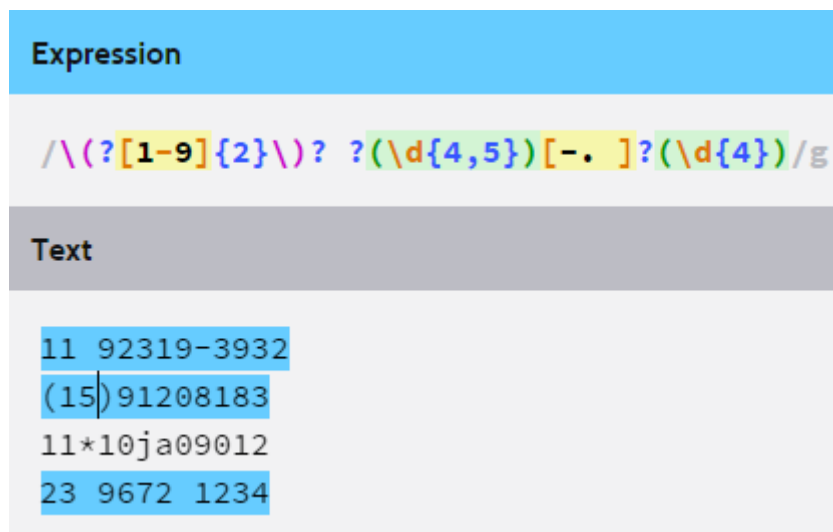


Figura 3.7 - Expressão Regular para telefone.

3.4. BANCO DE DADOS

Os bancos de dados de todas as aplicações foram desenvolvidos em SQL Server da Microsoft e a disciplina **Banco de Dados** instruiu os primeiros passos para desenvolver uma estrutura sólida e coerente, de acordo com as formas normais. A disciplina de **Segurança de Dados** também contribuiu para que as informações fossem transferidas de forma criptografada e segura de usuários não autorizados.

Com banco de dados é possível criar consultas específicas para retornar qualquer tipo de informação necessária pelo usuário. Tabelas podem ser relacionadas através das chaves primárias e estrangeiras. As chaves primárias representam uma identificação única para a informação e a chave estrangeira é a representação de uma informação de outra tabela. A *Figura 3.8* apresenta o fluxo de uma consulta e os elementos possíveis, como: colunas a serem exibidas, as tabelas a serem consultadas, as condições de filtros, agrupamentos e filtros após os resultados.

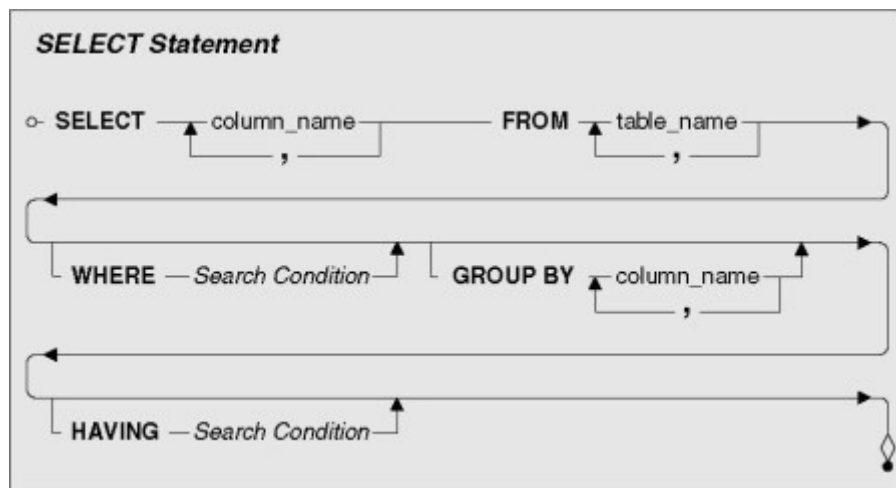


Figura 3.8 - Modelo de consulta em banco de dados.

A modelagem dos bancos desenvolvidos pode ser vista em *Figura 2.6* e *Figura 2.14*. Nelas é possível notar que estão nas formas normais, existem relacionamentos “um para um”, “um para n” e “n para n”.

Antes do desenvolvimento prático, foram feitas reuniões e modelagens conceituais para que fosse possível chegar ao modelo ideal sem que houvesse modificações durante o desenvolvimento das aplicações.

4. CONSIDERAÇÕES FINAIS

Este capítulo tem por objetivo descrever as contribuições que o estágio proporcionou à formação acadêmica e profissional, as dificuldades encontradas durante o processo e sugestões de trabalhos futuros.

4.1. CONTRIBUIÇÕES PARA A FORMAÇÃO

O estágio exigiu conhecimento e aplicação de várias disciplinas como Processamento da Informação, Programação para Web, Programação Orientada a Objetos, Banco de Dados, Lógica Básica, Computadores, Ética e Sociedade, Linguagens Formais e Autômata, Segurança de Dados, Programação para Dispositivos Móveis, Análise de Algoritmos e Engenharia de Software. O estágio exigiu o conhecimento para criar lógicas que solucionassem problemas, operadores lógicos, expressões regulares, consultas em banco de dados, desenvolvimento em camadas (MVC) e planejamento.

As ferramentas ensinadas ao aluno durante os cursos de formação facilitaram as atividades e exigiram uma curva de aprendizado menor nos assuntos em que foi necessário aprofundamento, considerando que o aprendizado em um ambiente acadêmico acontece em um cenário ideal e pequeno em relação ao mercado de trabalho.

O período total foi suficiente para perceber a diferença entre a teoria e a prática. Na teoria todo o assunto é passado ao aluno, os conceitos, maneiras de resoluções de problemas, os exercícios e projetos são grandes desafios e são os que põem a prova tudo o que foi aprendido nas aulas, porém, quando o aluno se depara com a realidade do mercado de trabalho, nota a diferença da dimensão de tudo o que foi visto antes e que nas aulas, apesar de parecer um projeto grande, é apenas um pequeno exemplo que pode acontecer na realidade. Essa distância entre a teoria e a prática é o que amadurece e faz com que o estagiário aplique tudo o que foi aprendido, contribuindo no crescimento profissional.

4.2. DIFICULDADES ENCONTRADAS

O desenvolvimento de aplicações no ambiente de produção exige muita disciplina do profissional. O código deve ser bem estruturado, pois será mantido por uma equipe. Deve haver

um planejamento do projeto antes de iniciar qualquer programação para evitar erros e trabalhos desnecessários. O profissional necessita pesquisar muito sobre como realizar a tarefa, qual seria a melhor solução, quais seriam as melhores ferramentas, conversar com outros desenvolvedores para adquirir experiência e criar uma solução ideal para o problema.

Algumas ferramentas utilizadas no decorrer do estágio não foram ensinadas nos cursos de formação do aluno, dificultando inicialmente o desenvolvimento dos projetos. Porém, tal fato implicou em um maior empenho e dedicação em estudos e pesquisas, gerando ótimos resultados nos projetos e amadurecendo o aluno profissionalmente.

4.3. SUGESTÕES DE TRABALHOS FUTUROS

As disciplinas ministradas aos alunos devem possuir uma reciclagem periodicamente, pois a área de tecnologia possui novidades constantemente. Além da reciclagem nos assuntos de cada disciplina, também é preciso verificar a necessidade de novas disciplinas que possam abranger as novidades de hardware e/ou software.

Os projetos desenvolvidos pelo aluno possuem tecnologias e ferramentas recentes e esse caminho deve ser mantido. Foram utilizadas ferramentas de código aberto, que facilita a atualização por parte dos desenvolvedores e da comunidade que participa com constantes contribuições. Outras ferramentas e recursos podem ser agregados aos sistemas para facilitar a experiência do usuário, mas com cuidado para que não haja exageros e conflitos com as ferramentas atuais.

O AngularJS demonstrou ser uma ferramenta com pequena curva aprendizado para tarefas mais comuns em um sistema. Ele deve ser mantido e aperfeiçoado nos sistemas atuais e cotado para o desenvolvimento de novos sistemas.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Tripletech, <http://www.tripletech.com.br/>, acessado em 03/08/2015 às 06h50.
- [2] Faculdade de Direito de São Bernardo do Campo, <http://www.direitosbc.br/>, acessado em 03/08/2015 às 07h00.
- [3] Faculdade de Direito de São Bernardo do Campo - Bolsas de Estudos, <http://www.direitosbc.br/a-faculdade-servicos-bolsa-de-estudo.aspx>, acessado em 04/08/2015 às 07h25.
- [4] AngularJS by Google, <https://angularjs.org/>, acessado em 04/08/2015 às 07h40.
- [5] JSON, <http://json.org/>, acessado em 04/08/2015 às 07h45.
- [6] Bootstrap, <http://getbootstrap.com/>, acessado em 04/08/2015 às 08h15.
- [7] Runrun.it, <http://runrun.it/>, acessado em 04/08/2015 às 13h20.
- [8] Angular Material Design by Google, <https://material.angularjs.org/latest/#/>, acessado em 05/08/2015 às 13h50.
- [9] Google, <http://google.com.br/>, acessado em 05/08/2015 às 13h55.
- [10] Android 5.0 Lollipop, https://www.android.com/intl/pt-BR_br/versions/lollipop-5-0/, acessado em 06/08/2015 às 07h35.
- [11] Especificações Material Design by Google, <https://www.google.com/design/spec/material-design/introduction.html>, acessado em 06/08/2015 às 07h40.
- [12] Google Inbox, <http://inbox.google.com.br/>, acessado em 06/08/2015 às 07h45.
- [13] Google Music, <http://music.google.com.br/>, acessado em 06/08/2015 às 07h46.
- [14] Google Keep, <http://keep.google.com.br/>, acessado em 06/08/2015 às 07h47.
- [15] Sommerville, I. Engenharia de Software. 8.ed. - São Paulo : Addison-Wesley, 2007.
- [16] Pressman, Roger S. Engenharia de Software. 6.ed. - Rio de Janeiro: McGraw-Hill, 2006
- [17] O que é DOM, <http://tableless.com.br/tenha-o-dom/>, Acessado em 06/08/2015 às 13h35