

DAS Departamento de Automação e Sistemas
CTC **Centro Tecnológico**
UFSC Universidade Federal de Santa Catarina

Desenvolvimento de uma Plataforma Web para Gestão de Relacionamento com o Cliente

*Relatório submetido à Universidade Federal de Santa Catarina
como requisito para a aprovação da disciplina:
DAS 5511: Projeto de Fim de Curso*

Matheus Felipe Souza Valin

Florianópolis, Novembro de 2018

Desenvolvimento de uma Plataforma Web para Gestão de Relacionamento com o Cliente

Matheus Felipe Souza Valin

Esta monografia foi julgada no contexto da disciplina

DAS 5511: Projeto de Fim de Curso

e aprovada na sua forma final pelo

Curso de Engenharia de Controle e Automação

Prof. Rômulo Silva de Oliveira

Banca Examinadora:

Prof. Rômulo Silva de Oliveira
Orientador no Curso

Prof. Julio Elias Normey Rico
Responsável pela disciplina

Rafhael Cunha, Avaliador

Ígor Assis Rocha Yamamoto , Debatedor

Rafael Scheffer, Debatedor

Agradecimentos

Agradeço, a Deus, por toda essa jornada.

Aos meus pais, João Weber Valin e Janete Souza Valin, por tudo o que me ensinaram e pelo enorme apoio durante toda a graduação.

À minha namorada e aos meus amigos por todo o apoio acadêmico e emocional, sem os quais eu não estaria onde estou.

Aos professores da Universidade Federal de Santa Catarina, especialmente ao meu orientador, professor Rômulo Silva de Oliveira, pela disposição e apoio ao desenvolvimento deste projeto.

Ao Centro Acadêmico de Eng. De Controle e Automação, pelo crescimento pessoal que me proporcionou.

À equipe Emusys, por todo o apoio e auxílio criativo no desenvolvimento deste projeto.

Resumo

A ValinSW é uma jovem empresa de desenvolvimento de software, cujo principal produto é o sistema Emusys, uma plataforma web de gestão de escolas de música, que tem como objetivo ser o mais completo e flexível possível, ajustando-se às necessidades e particularidades de cada escola cliente da empresa.

Conforme o número de clientes da empresa cresce, as atividades internas essenciais da empresa, como o atendimento e suporte a seus clientes, tem se tornado cada vez mais difíceis e por vezes deficientes devido à falta de uma organização e registro de informações. Dessa forma, o foco deste trabalho é o desenvolvimento de uma plataforma web que contemple e auxilie em todos os principais processos internos da empresa, tornando-os melhores, mais ágeis e mais escaláveis.

A plataforma desenvolvida permite o registro, de forma ágil e rápida, de todos os contatos realizados entre a empresa e seus clientes, assim como a consulta e análise desses dados, auxiliando e melhorando o atendimento aos clientes da empresa. A Plataforma permite também o registro de pendências da empresa para com seus clientes, organizando estas pendências e auxiliando colaboradores da empresa no cumprimento das mesmas.

Palavras-chave: Plataforma, Web, Relacionamento com o cliente

Abstract

ValinSW is a young software development company, whose main product is the Emusys system, a managing web platform for schools of music, which aims to be the most complete and flexible system possible, adjusting itself to the needs and particularities of each school of music, client of this company.

As the number of company clients grow, the internal activities of the company, such as client support and issues registering, has increasing difficulties and many times deficiencies due to a faulting organization and information registration.

The platform developed in this paper allows, in a fast and agile way, the registration of all communication between the company and its clients, as well as the consulting and analyzing of this data, therefore helping and improving client support. The platform allows as well the registration of company pendencies, organizing these pendencies and allowing company colaborators in the finishing of those.

Keywords:Platform, Web, Client Relationship

Lista de ilustrações

Figura 1 – Tela Principal do Emusys, escola de teste, dados fictícios.	20
Figura 2 – Agenda do Emusys, escola de teste, dados fictícios.	20
Figura 3 – O Sistema Manager, Painel Financeiro	21
Figura 4 – Modelo Incremental - Extraído de Sommerville [1]	24
Figura 5 – Diagrama Geral do funcionamento do <i>framework</i> Emusys	25
Figura 6 – Código exemplo de uso do Framework Emusys, para criação de uma janela modal (<i>dialog</i>) mostrada na figura 7	26
Figura 7 – Resultado do código da figura 6	27
Figura 8 – Diagrama de Entidades	33
Figura 9 – O PhpStorm	35
Figura 10 – Diagrama da tabela "contato"e suas tabelas relativas	37
Figura 11 – Diagrama da tabela "ticket"e suas tabelas relativas	38
Figura 12 – Diagrama com cenários simples de organização de fila de tickets	40
Figura 13 – A Página Escola	42
Figura 14 – Lista de Contatos da Página Escola	43
Figura 15 – Lista de tickets associados a um contato	44
Figura 16 – Janela de registro de um novo Ticket, a partir de um contato	46
Figura 17 – A Página Ticket	47
Figura 18 – Exemplo de anotações, históricos e contatos registrados em um ticket	49
Figura 19 – A Página Pendências	50
Figura 20 – Destaque da lista de Pendências	51
Figura 21 – A Página "Meus Ticket- Seção Painel	52
Figura 22 – A Página "Meus Ticket- Seção Fila	53
Figura 23 – A Página "Meus Tickets- Seção Notificações	54

Lista de tabelas

Lista de abreviaturas e siglas

SGBD - Sistema de Gerenciamento de Banco de Dados

IDE - *Integrated Development Environment* (Ambiente de Desenvolvimento Integrado)

JSON - *Javascript Object Notation* (Notação de Objetos Javascript)

SQL - *Structured Query Language* (Linguagem de Consulta Estruturada)

UFSC - Universidade Federal de Santa Catarina

PFC - Projeto de Fim de Curso

Sumário

1	INTRODUÇÃO	17
1.1	Contextualização	17
1.2	Objetivos Gerais	17
1.3	Objetivos Específicos	17
1.4	Delimitação do Projeto	18
1.5	Organização do Documento	18
2	A EMPRESA	19
2.1	A ValinSW	19
2.2	O Emusys	19
2.3	O Sistema Manager	21
3	FUNDAMENTAÇÃO TEÓRICA E FERRAMENTAS	23
3.1	Modelo Incremental de Desenvolvimento	23
3.2	Linguagem de Programação - PHP	23
3.3	O SGBD - MariaDB	24
3.4	O Framework Emusys	24
4	MODELAGEM DA PLATAFORMA	29
4.1	Requisitos de Usuário	29
4.2	Requisitos Não-Funcionais	30
4.3	Modelagem dos Dados	30
4.3.1	Contato	31
4.3.2	Ticket	31
4.3.3	A Entidade Escola	33
5	DESENVOLVIMENTO DA PLATAFORMA	35
5.1	Escolha da IDE	35
5.2	Banco de Dados	36
5.2.1	A Tabela Contato	36
5.2.2	A Tabela Ticket	36
5.3	O Sistema de Estimativas	38
5.4	Interface	41
5.4.1	A Página Escola	41
5.4.1.1	Registro de Contato	42
5.4.1.2	Lista de Contatos	42

5.4.1.3	Registro de Ticket	44
5.4.2	A Página Ticket	46
5.4.2.1	A Coluna de Informações Gerais	47
5.4.2.2	Botões de Ação Rápida	48
5.4.2.3	Registro de Anotações e Menções	48
5.4.3	A Página Pendências	49
5.4.3.1	A Tabela de Tickets Pendentes	50
5.4.4	A Página de Tickets Delegados	51
5.4.4.1	Em Andamento	52
5.4.4.2	Fila Pessoal	53
5.4.4.3	Notificações	53
6	ANÁLISE DOS RESULTADOS	55
7	CONCLUSÕES E PERSPECTIVAS	57
	REFERÊNCIAS	59

1 Introdução

1.1 Contextualização

Atualmente, com o avanço das mídias sociais, o relacionamento com o cliente tem tido cada vez mais foco pelas empresas, que utilizam da tecnologia para criarem experiências de atendimento cada vez melhores, mais dedicadas e personalizadas.

Na ValinSW, o relacionamento com o cliente é e sempre foi um grande foco. Durante o período inicial da empresa, devido ao baixo número de clientes, o atendimento era realizado por uma única pessoa, que conhecia bem todos os clientes, suas necessidades e particularidades, e assim o atendimento era pessoal, amigável e altamente efetivo.

Porém, com o crescimento do número de clientes, e o conseqüente crescimento do número de colaboradores, o processo de atendimento ao cliente da empresa foi se fragmentando, e logo se tornou evidente a necessidade de um registro adequado das informações pertinentes de cada cliente, assim como um registro dos contatos realizados e das pendências para com cada cliente.

Assim, surgiu a iniciativa do projeto objeto deste documento: um software que auxiliasse a empresa em suas principais atividades de relacionamento com o cliente, criado sob medida para a forma com que a empresa funciona atualmente.

1.2 Objetivos Gerais

O principal objetivo deste projeto de fim de curso é o desenvolvimento e a implementação de uma plataforma web de gestão de relacionamento com o cliente, que, utilizado pelos colaboradores da empresa, auxilie as atividades de atendimento, suporte e a prospecção de novos clientes.

1.3 Objetivos Específicos

- Estudar os processos de relacionamento com o cliente da empresa;
- Modelar a plataforma e sua usabilidade;
- Implementar a plataforma no dia a dia da empresa;
- Melhorar a plataforma a partir de *feedback* interno.

1.4 Delimitação do Projeto

O projeto deste documento envolve o planejamento da plataforma e de seu uso dentro da empresa, e o desenvolvimento de todas as suas partes, incluindo o banco de dados, interfaces gráficas de uso do usuário e lógicas e regras internas. O Desenvolvimento deste projeto foi inteiramente realizado pelo aluno, com a equipe da empresa ValinSW auxiliando na modelagem das especificações do projeto e em tomadas de decisão com relação a interfaces e usabilidade apenas.

1.5 Organização do Documento

- **A Empresa:** - Apresentação da empresa ValinSW.
- **Fundamentação Teórica e Ferramentas:** - Apresentação da metodologia de desenvolvimento utilizada, e as principais ferramentas utilizadas no desenvolvimento da plataforma.
- **Modelagem da Plataforma:** - Apresentação da modelagem conceitual da plataforma, incluindo requisitos funcionais e não funcionais e modelagem de dados.
- **Desenvolvimento da Plataforma:** - Neste capítulo, é apresentada a plataforma desenvolvida, incluindo o banco de dados, sistemas auxiliares e o design das principais interfaces.
- **Análise dos Resultados:** - Abordagem dos resultados do projeto, seu uso na empresa e cumprimento dos requisitos.
- **Conclusões e Perspectivas:** - Conclusão do projeto e perspectivas futuras.

2 A Empresa

2.1 A ValinSW

A ValinSW é uma empresa familiar de desenvolvimento de software, fundada em 2016, que tem como principal produto o sistema Emusys, uma plataforma web de gestão para escolas de música.

A ValinSW foi fundada com o objetivo de desenvolver o melhor sistema de gestão para escola de música, focando exclusivamente neste pequeno nicho e desenvolvendo desde o início um sistema sob medida para as necessidades das escolas de música do Brasil, desde as menores escolas de bairro até os grandes conservatórios.

Dessa forma, a empresa possui um relacionamento dedicado com cada cliente, de modo a conhecer os processos e atender as necessidades particulares de cada um. Em termos práticos, isso significa que a plataforma Emusys está sob constante modificação e melhoria, com novas funcionalidades e adaptações sendo adicionadas semanalmente.

2.2 O Emusys

O Emusys é uma plataforma online, uma aplicação executada em navegadores e, portanto, pode ser utilizada de qualquer lugar em qualquer computador, sem a necessidade de instalação. O sistema é comercializado através do modelo SaaS (*Software as a Service*), ou seja, o pagamento é mensal através de assinatura e a escola pode utilizar o sistema enquanto o pagamento for realizado, não havendo necessidade de comprar licenças. A tela principal do Emusys e a tela de agenda de aulas podem ser visualizadas respectivamente nas figuras 1 e 2.

O sistema fornece as seguintes funcionalidades para escolas de música:

- Registro e organização de cursos, módulos e disciplinas de instrumentos;
- Gestão de matrículas, trancamentos, renovações, inadimplência;
- Gestão de turmas;
- Agendamento de aulas, incluindo reposições, reagendamentos e controle de presença;
- Cálculo e controle de pagamento de professores;
- Controle Financeiro interno da empresa;

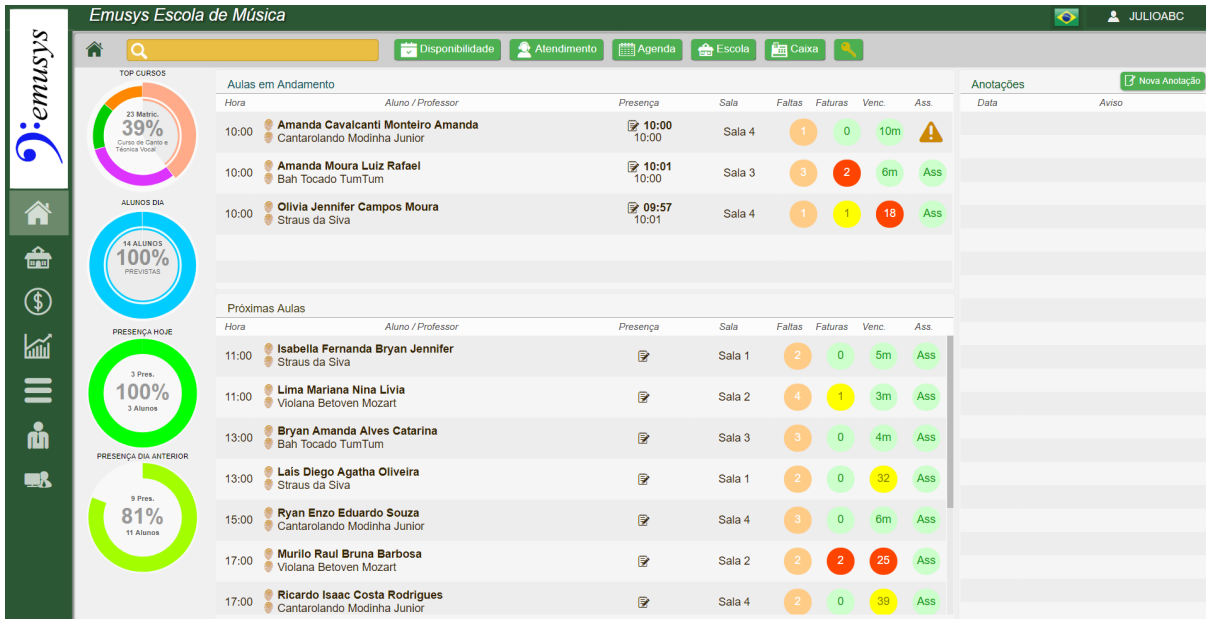


Figura 1 – Tela Principal do Emusys, escola de teste, dados fictícios.

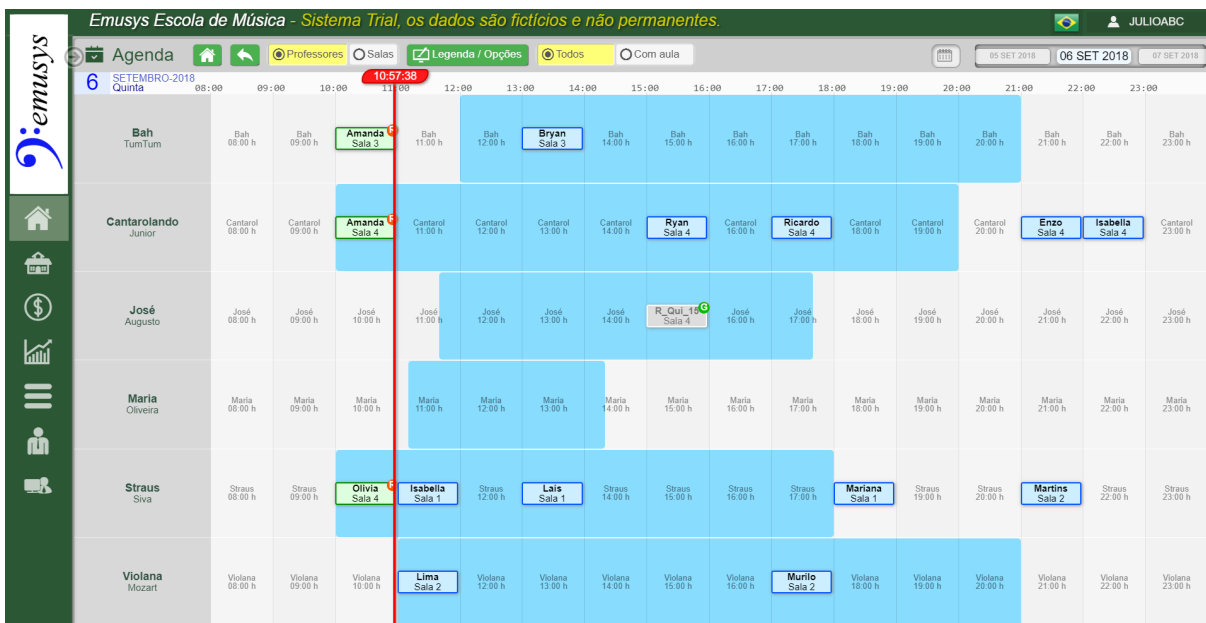


Figura 2 – Agenda do Emusys, escola de teste, dados fictícios.

- Gestão de disponibilidade de horários, professores e salas.

Além do sistema web, o Emusys também inclui aplicativos móveis para celulares Android e iPhone, voltados para alunos e para professores das escolas, contando com funcionalidades como confirmação de aulas, reposições, acompanhamento de horários, entre outras.

2.3 O Sistema Manager

Para controle interno da empresa, foi desenvolvido também o sistema Manager, uma plataforma por onde a administração da empresa pode habilitar ou desabilitar clientes, acompanhar o uso da plataforma e registrar pagamentos de mensalidades de cada escola. O Manager também contém o mesmo módulo de controle financeiro do Emusys, onde as finanças da empresa são organizadas e acompanhadas. Devido ao ambiente já organizado e à facilidade de acesso, a plataforma objeto deste documento foi desenvolvida como um módulo a mais no sistema Manager, onde ela poderá ser acessada pelos colaboradores da empresa de forma natural. O painel financeiro do sistema Manager pode ser visualizado na figura 3



Figura 3 – O Sistema Manager, Painel Financeiro

3 Fundamentação Teórica e Ferramentas

3.1 Modelo Incremental de Desenvolvimento

O objetivo final da plataforma desenvolvida neste projeto é fazer parte dos processos internos da empresa, ou seja, ser utilizado diariamente pelos colaboradores da empresa, de modo a auxiliar e facilitar tarefas e processos. Por este motivo, a plataforma precisa ser modelada de acordo com as atividades da empresa e com a forma com que estas são realizadas, de modo a não se tornar um processo complicado ou trabalhoso para as partes envolvidas, e sim apenas auxiliar e facilitar. Devido a este objetivo, e também a necessidade de uma implementação em curto prazo para que a empresa seja auxiliada o mais rápido possível, foi optado pelo modelo Incremental de desenvolvimento.

O Modelo Incremental de desenvolvimento é um método de desenvolvimento de software descrito por Sommerville (2011) [1] no qual se desenvolve primeiramente uma versão inicial do software, capaz de realizar as tarefas básicas a que o software se propõe, e já implementá-la e apresentá-la ao usuário. Assim, conforme o usuário for utilizando este novo software, suas opiniões e ideias, ou seja, seu *feedback*, é estudado e implementado no sistema em forma de melhorias e novas funcionalidades, em versões posteriores. Um esquema desta metodologia pode ser visto na figura 4.

Desta forma, a partir da versão inicial do software, pode-se implementar a plataforma, ao menos parcialmente, na realização de atividades da empresa e, com isso, conseguir não só um *feedback* em relação à plataforma em si, mas também adquirir perspectivas sobre como o uso da plataforma poderá impactar as atividades da empresa e conseqüentemente ajustar melhor o software para adequar a estes impactos, de forma contínua e iterativa.

3.2 Linguagem de Programação - PHP

A linguagem de programação utilizada para desenvolvimento da plataforma foi o PHP, uma vez que esta já é a linguagem primariamente utilizada na empresa e também foi utilizada para desenvolver o sistema Manager, o qual será utilizado como base para o desenvolvimento da plataforma.

O PHP (PHP: *Hypertext Processor*) é uma linguagem de programação *open-source* especialmente adequada para o desenvolvimento de aplicações de servidores para a web [2]. O PHP foi escolhido para o desenvolvimento do produto desta empresa por ser uma linguagem que proporciona um desenvolvimento rápido e prático, além de ser estável e ser utilizada por alguns dos sites de maior volume de acessos da atualidade [3].

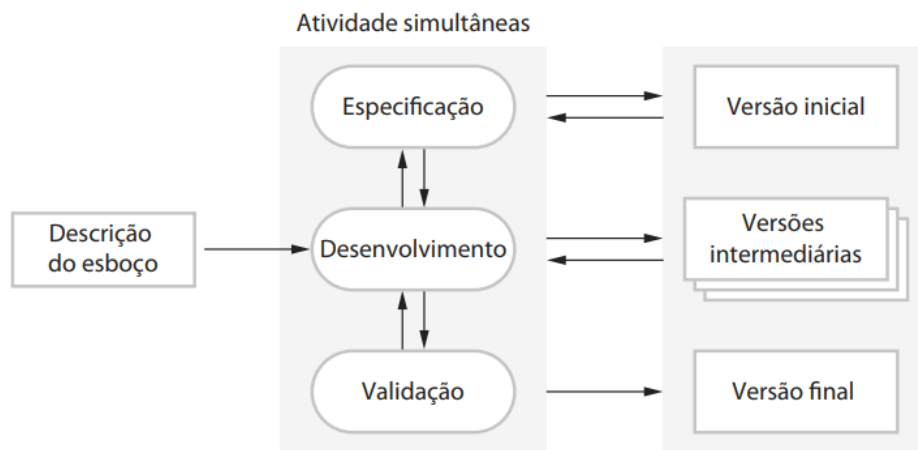


Figura 4 – Modelo Incremental - Extraído de Sommerville [1]

3.3 O SGBD - MariaDB

O SGBD (Sistema de gerenciamento de banco de dados) utilizado para o desenvolvimento desta plataforma foi o MariaDB, um sistema *open-source* derivado do tradicional SGBD MySQL [4], o qual hoje possui uma grande parcela do mercado de sistemas de gerenciamento de banco de dados [5]. O MariaDB é um sistema de gerenciamento de banco de dados relacional, ou seja, trabalha com relações entre dados armazenados em tabelas.

O uso do MariaDB, da mesma forma que a linguagem de programação, foi adotado principalmente pelo uso já corrente desta tecnologia na empresa.

3.4 O Framework Emusys

Desde a fundação da empresa, o sistema Emusys foi desenvolvido com utilização de um *framework* próprio, criado especialmente para o desenvolvimento do sistema. Criado a partir de uma filosofia centralizada, o *framework* é um grupo de classes e funcionalidades que permitem que o desenvolvimento completo de um sistema, inclusive de suas interfaces e design de telas, seja feito unicamente pela linguagem de *back-end* (PHP), sem necessariamente a escrita de HTML [6] ou JavaScript [7], tão comum neste tipo de aplicação. Isso é possível pois o *framework* se baseia em uma aplicação para navegadores que recebe apenas instruções e dados, e, a partir destas informações, gera a interface a ser mostrada para o usuário. Assim, não há tráfego de HTML entre servidor e navegador, apenas tráfego de informações organizadas utilizando o formato JSON [8]. Um diagrama geral do funcionamento deste *framework* pode ser visualizado na figura 5. Um exemplo do uso deste *framework* para a criação de um simples formulário em uma janela modal, chamada de *dialog*, pode ser visualizado nas figuras 6 e 7. O código mostrado na figura

6 gera um conjunto de informações no formato JSON que, enviado para o navegador, é utilizado pela aplicação JavaScript para montar a janela mostrada na figura 7.

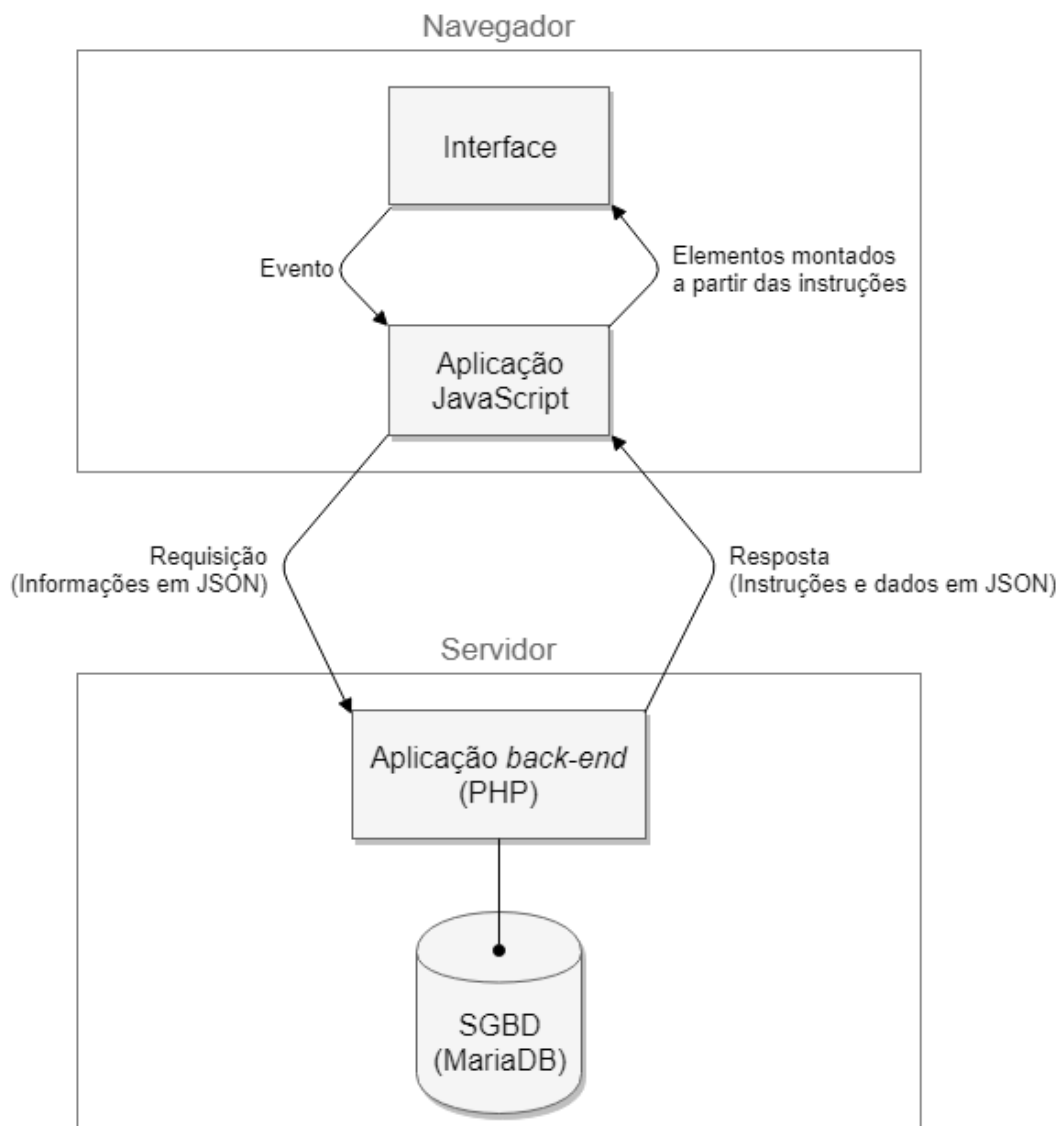


Figura 5 – Diagrama Geral do funcionamento do *framework* Emusys

A utilização de um *framework* próprio tem suas vantagens e desvantagens. A principal vantagem é a de que se pode criar o que quiser, se o *framework* não for capaz de algo, é só modificá-lo para que seja, sem depender de terceiros. Esta vantagem é também considerada uma desvantagem, uma vez que tudo que se gostaria de ter em um sistema, deve se implementar. Porém, com o tempo, o *framework* Emusys foi crescendo e se aperfeiçoando, inclusive por ser utilizado tanto no produto da empresa quanto nas

ferramentas que a própria empresa depende para funcionar. Hoje em dia, os colaboradores que trabalham no sistema trabalham também neste *framework* conforme necessidade.

Durante o desenvolvimento deste projeto, foram desenvolvidas também modificações e melhorias para o framework, para serem utilizadas na plataforma.

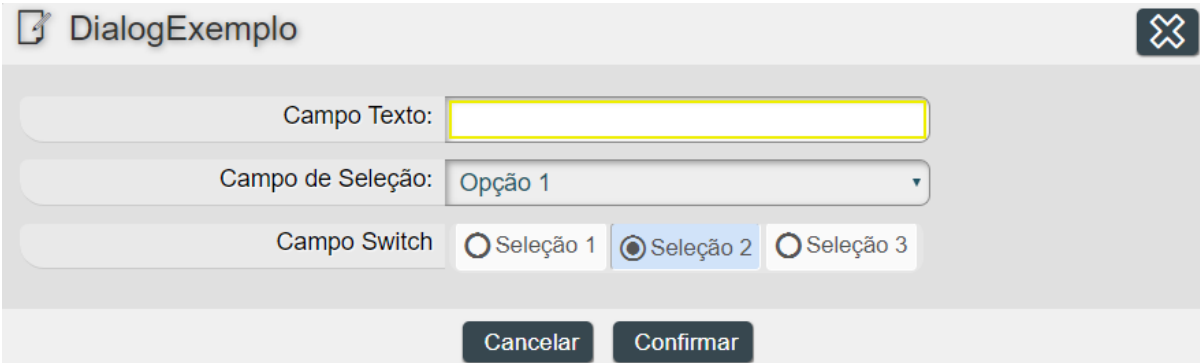
```
$dialog = ZDialog::new( titulo: "DialogExemplo", icon: ZIcon::New );

$dialog->getBodyBox()->addItens(
  ZTextInput::new( id: 'id1', label: "Campo Texto:", width: 20),
  ZSelect::new2( id: 'id2', label: 'Campo de Seleção:', width: 20)
  ->setValueFromPOST()
  ->addItens(
    ZKeyValue::new(1, 'Opção 1'),
    ZKeyValue::new(2, 'Opção 2')
  ),
  ZSwitch::new( id: 'id3', label: 'Campo Switch')
  ->setValueFromPOST()
  ->addItens(
    ZKeyValue::new(1, 'Seleção 1'),
    ZKeyValue::new(2, 'Seleção 2'),
    ZKeyValue::new(3, 'Seleção 3')
  )
);

$dialog->getFooterBox()->addItens(
  ZButton::new( text: "Cancelar", ZAction::fechar()),
  ZButton::new( text: "Confirmar", ZAction::refresh())
);

$dialog->show();
```

Figura 6 – Código exemplo de uso do Framework Emusys, para criação de uma janela modal (*dialog*) mostrada na figura 7



The image shows a dialog box titled "DialogExemplo" with a close button in the top right corner. It contains three input fields:

- Campo Texto:** A text input field with a yellow border.
- Campo de Seleção:** A dropdown menu showing "Opção 1".
- Campo Switch:** Three radio button options: "Seleção 1", "Seleção 2" (which is selected and highlighted in blue), and "Seleção 3".

At the bottom of the dialog are two buttons: "Cancelar" and "Confirmar".

Figura 7 – Resultado do código da figura 6

4 Modelagem da Plataforma

Para a modelagem e planejamento da plataforma a ser desenvolvida, começa-se pela definição dos requisitos do software. De acordo com Sommerville (2011) [1], "Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento. Esses requisitos refletem as necessidades dos clientes para um sistema que serve a uma finalidade determinada, como controlar um dispositivo, colocar um pedido ou encontrar informações".

No caso da plataforma objeto deste documento, os clientes são os próprios colaboradores da empresa. Dessa forma, os requisitos a seguir foram levantados através de observação do dia a dia destes colaboradores no desenvolvimento de suas funções ligadas ao objetivo desta plataforma, assim como através de conversas com os mesmos e os diretores da empresa.

4.1 Requisitos de Usuário

Os requisitos de usuário dizem respeito a que serviços a plataforma fornecerá a seus usuários finais. Nos itens a seguir, a palavra "cliente" diz respeito a uma empresa (escola de música) cliente da ValinSW.

- Um usuário deve poder registrar um contato com um cliente, incluindo o que foi falado;
- Um usuário deve poder registrar pendências da empresa com relação a seus clientes;
- Um usuário deve poder visualizar facilmente todas as comunicações realizadas com um dado cliente;
- Um usuário deve poder visualizar facilmente todas as pendências existentes para com um dado cliente;
- Um usuário deve ser capaz de registrar anotações associadas a pendências existentes para com um dado cliente;
- Um usuário deve poder registrar, em uma dada pendência, seu status atual com relação a seu cumprimento (pendente, em desenvolvimento, finalizado);
- Um usuário deve ser capaz de registrar, uma dada pendência, outro usuário ou a si mesmo como responsável pela pendência;

- Um usuário deve ser capaz de visualizar facilmente todas as pendências das quais ele é responsável;
- Um usuário deve ser capaz de comunicar-se com outros usuários através de mensagens.

4.2 Requisitos Não-Funcionais

Requisitos não funcionais de um sistema são, conforme Sommerville [1], "Os requisitos não funcionais, como o nome sugere, são requisitos que não estão diretamente relacionados com os serviços específicos oferecidos pelo sistema a seus usuários. Eles podem estar relacionados às propriedades emergentes do sistema, como confiabilidade, tempo de resposta e ocupação de área".

No caso desta plataforma, uma vez que a ideia é que esta seja utilizada diariamente nas atividades da empresa, confiabilidade e estabilidade são essenciais. Porém, além destes requisitos, grande atenção foi dada à questão da facilidade de uso e rapidez da plataforma, uma vez que um sistema de registro de informações que seja difícil ou trabalhoso de usar acaba sendo um fardo na jornada de trabalho de colaboradores, e poderia acabar não sendo utilizado de forma adequada. Em determinados momentos do dia, há grande procura de atendimento por parte dos clientes, o que pode acabar sobrecarregando atendentes. Em situações como esta, se o software de registro destes atendimentos for lento ou trabalhoso para se usar, isso rapidamente se torna inviável. Assim, foram definidas os seguintes requisitos não funcionais:

- A plataforma deve ser de fácil e prática operação;
- A plataforma não deve apresentar erros que atrapalhem o seu uso corriqueiro;
- A plataforma deve ser rápida, com todo e qualquer tempo de espera sendo menor que meio segundo;
- A plataforma não deverá jamais perder dados registrados nela.

4.3 Modelagem dos Dados

Uma vez que esta plataforma é primariamente um sistema de registro e consulta de informações, a modelagem se iniciou pelo detalhamento das estruturas de dados da plataforma. Através de estudo dos requisitos definidos na seção anterior, foi definido que a plataforma teria duas entidades fundamentais, o "Contato" e o "Ticket".

4.3.1 Contato

A entidade contato foi concebida para registrar toda e qualquer comunicação realizada entre a empresa e seus clientes. A partir de conversas com colaboradores e diretores, e observações do processo de contato e das informações necessárias para posterior análise, foram definidas as seguintes informações como partes constituintes da entidade Contato.

- Com qual empresa foi o contato;
- Qual o nome do cliente (pessoa) foi feito o contato;
- Qual foi a forma de contato (Ligação, E-mail, Aplicativo de Mensagens);
- O que foi falado neste contato;
- Qual colaborador da empresa falou com o cliente;
- Qual a data e a hora em que o contato foi realizado.

Com o registro destes dados em cada contato, é possível realizar consultas posteriores como, por exemplo, todos os contatos realizados com um dado cliente, ou em um certo dia.

A partir desta entidade, foi considerado que muitas vezes um atendimento que a empresa realiza não é encerrado quando o contato acaba. Muitas vezes, devido à proposta da empresa ValinSW conforme especificado no capítulo 2, os clientes requisitam uma modificação ou uma nova funcionalidade para o sistema, de forma que surge uma pendência, a partir do contato, que deve ser resolvida pelos desenvolvedores da empresa. Considerou-se que estas pendências também devem ser registradas, e assim criou-se a entidade Ticket

4.3.2 Ticket

A entidade Ticket representa uma pendência da empresa para um ou mais clientes, e nasce a partir de um contato realizado entre a empresa e seu cliente. Esta pendência eventualmente se tornará responsabilidade de um desenvolvedor, colaborador da empresa, e este desenvolvedor trabalhará nela e eventualmente a declarará como finalizada. Dessa forma, é necessário que cada pendência tenha um status, classificando-a de acordo com seu estágio no ciclo de vida: Pendente, Em Andamento, Em Espera e Finalizado.

Assim, chegou-se na seguinte lista de informações que uma pendência da empresa, ou seja, um Ticket, deveria conter:

- Título do ticket, para fácil identificação e listagem;
- Descrição do Ticket;

- Qual contato deu origem a este ticket (Em caso de mais de uma escola requisitando a mesma funcionalidade, o Ticket deve ter essa relação com todos os contatos associados);
- Qual colaborador da empresa registrou este ticket;
- Plataforma do ticket (é algo relacionado ao Sistema, ao aplicativo para *Android* ou ao aplicativo para *iPhone*);
- Deadline (não é incomum um cliente requisitar uma previsão para quando algo que ele requisitou será entregue. Assim, é necessário registrar a data passada para o cliente);
- Status do Ticket (Em Aberto, Em Andamento, Em Espera, Aguardando Deploy, Finalizado);
- Qual desenvolvedor está responsável por este ticket;
- Qual atendente está responsável por este ticket (Em geral seria o próprio atendente que registrou o ticket. Esta informação é útil para que o desenvolvedor do ticket saiba com quem falar em caso de dúvidas ou de necessidade de contato adicional sobre aquele ticket com o cliente).

Além das informações básicas, listadas acima, também considerou-se a necessidade de adicionar outras informações, não essenciais, mas que seriam úteis para classificar os tickets, organizar e criar estatísticas a partir de dados passados. Assim, os tickets também conteriam:

- Tempo Estimado - Classificação de tempo necessário para resolver aquela pendência (em termos de horas, dias ou semanas). Tal dado seria informado pelos desenvolvedores ao estudar do que se trata o ticket, e pode ser utilizado em termos de priorização de tickets ou cumprimento de *deadlines*.
- Tipo do Ticket - Classificação do tipo do ticket, se trata-se de um problema ou "bug", de uma funcionalidade nova que o cliente requer no sistema, de uma requisição de configuração ou modificação não programática no sistema, etc.
- Nível do Ticket - Classificação do nível de prioridade do ticket, entre Urgente e não urgente.

Também decidiu-se que cada ticket deveria possuir um espaço para anotações, de modo que desenvolvedores e atendentes da empresa poderiam anotar conforme houvessem modificações naquela requisição, progresso em seu desenvolvimento, necessidade de mais informações por parte do cliente, entre outras situações semelhantes.

Por fim, considerou-se bom registrar cada passagem do ticket de um status para outro (ou seja, de Em aberto para Em Andamento e assim por diante), no tempo, de forma que se pudesse posteriormente criar estatísticas sobre quanto tempo um ticket passou em cada status. Tal informação poderá ser útil futuramente.

4.3.3 A Entidade Escola

Para centralizar as duas entidades supracitadas, há uma entidade Escola, representando cada cliente que a empresa tem (Escolas de Música). Assim, cada escola terá contatos associados a ela, e cada contato poderá ter nenhum ou mais tickets associados a ele. Um ticket poderá ser associado a mais de um contato, inclusive podendo ser contatos de diferentes escolas, caso mais de uma escola requisite a mesma funcionalidade para o sistema, por exemplo. Um esquema geral da relação entre as entidades pode ser visto na figura 8

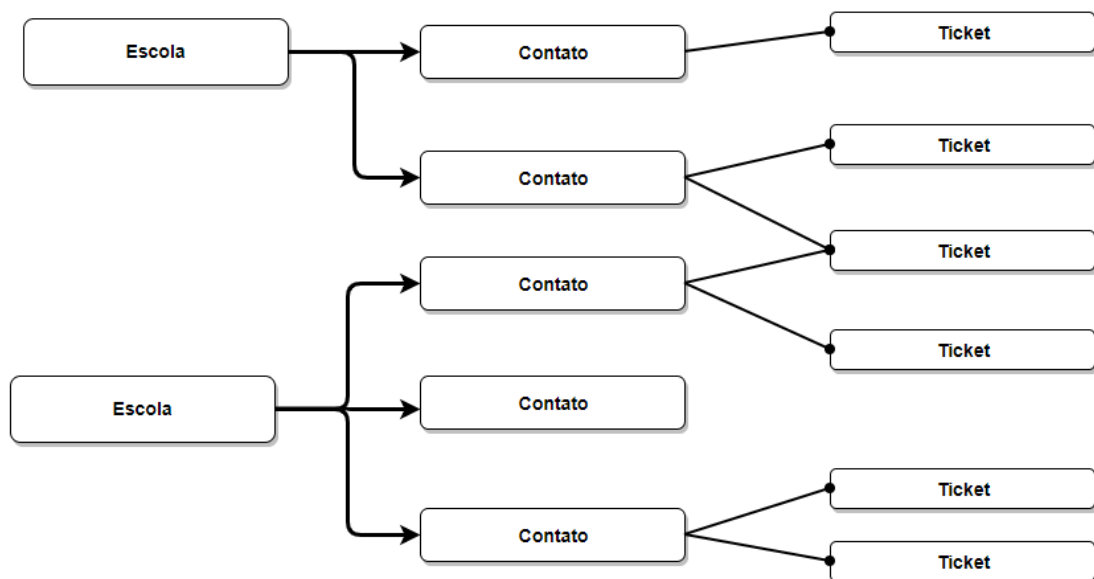


Figura 8 – Diagrama de Entidades

5 Desenvolvimento da Plataforma

Neste capítulo, é apresentado o processo de desenvolvimento da plataforma, incluindo design de telas, especificação do banco de dados e algoritmos. Uma vez que este desenvolvimento teve como foco a usabilidade do usuário, praticidade e fluidez de operação, a plataforma será apresentada através de suas interfaces. As etapas descritas a seguir foram desenvolvidas em sua totalidade pelo aluno.

5.1 Escolha da IDE

Para o desenvolvimento da plataforma, a IDE (*Integrated Development Environment*, Ambiente de Desenvolvimento Integrado) escolhida foi o PhpStorm, desenvolvida pela empresa JetBrains, por ser a principal IDE utilizada na empresa ValinSW e pelo aluno já estar familiarizado com ela. O PhpStorm é uma IDE completa para desenvolvimento web, contando com suporte não só para a linguagem PHP mas também para Javascript, HTML, CSS e SQL. Além disso, ela contém um suporte robusto para visualização, edição e consulta a banco de dados através do módulo para banco de dados presente na IDE, o que foi grandemente utilizado pelo aluno no decorrer deste projeto. A IDE pode ser visualizada na figura 9.

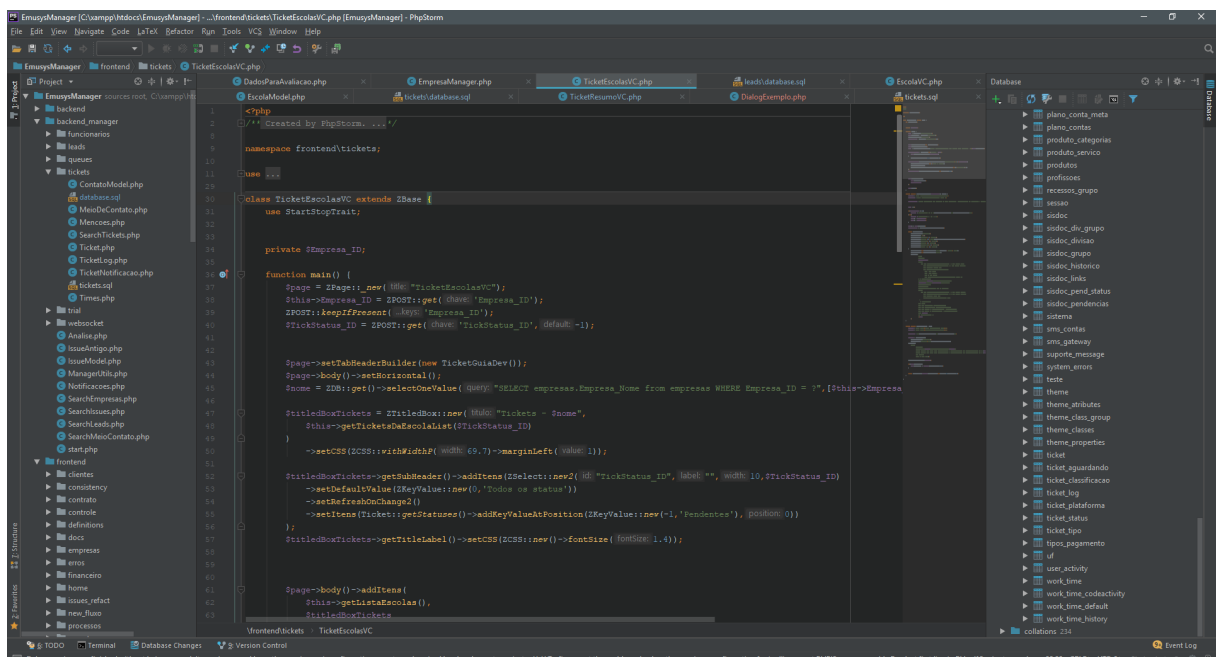


Figura 9 – O PhpStorm

5.2 Banco de Dados

A criação do banco de dados da plataforma se deu a partir das especificações da estrutura de informação do projeto, conforme detalhado no capítulo 4. Foram criadas tabelas para representar cada uma das entidades definidas na especificação do projeto. A seguir tem-se uma descrição das principais tabelas do sistema.

5.2.1 A Tabela Contato

Para a entidade Contato, foi criada a tabela de mesmo nome, e organizada de forma a conter todas as informações de forma relacional, aproveitando algumas informações já presentes em outras tabelas do sistema Manager. Um esquema da tabela Contato e suas relações pode ser visualizado na figura 10.

A Empresa que entrou em contato é registrada a partir do campo "Empresa_ID", o qual é uma chave estrangeira, referenciando a tabela de empresas já existente no banco de dados da empresa, com todas as informações pertinentes a cada empresa. O mesmo foi feito para a identificação do colaborador da empresa que realizou o atendimento, através do campo "Pessoa_ID_Atendente", também uma chave estrangeira que referencia a tabela pessoas, contendo registros de todos os colaboradores da empresa.

Para o registro do cliente (pessoa) que entrou em contato, dois campos foram utilizados. O campo "EmpUsu_ID" é uma chave estrangeira, ligando a tabela com a tabela de usuários do sistema, que contém todos os usuários do sistema, de todas as empresas. Desta forma, em vez de digitar o nome da pessoa com quem se falou, o atendente pode selecionar entre uma lista de nomes de usuários daquela empresa, acelerando o registro. Porém, há também a possibilidade, apesar de mínima, de se ter um contato com alguém que não é usuário do sistema. Devido a esta possibilidade, foi acrescentado também o campo "Contato_ClienteNome", no qual pode-se registrar o nome da pessoa, caso este não esteja na lista de usuários daquela empresa.

Os campos "Contato_DataHora" e "Contato_Descricao" são autoexplicativos, contendo a data e a hora em que o contato foi realizado e o que foi falado, respectivamente. O Campo Meio_ID foi criado para a informação de qual foi a forma de contato utilizada (Ligação, E-mail, Aplicativo de Mensagens), registrando os números 1, 2 e 3 para cada uma destas formas de contato, respectivamente.

5.2.2 A Tabela Ticket

A tabela ticket foi criada para registro da entidade Ticket, conforme definida na especificação do projeto. Um diagrama desta tabela e suas tabelas relativas pode ser visualizado na figura 11

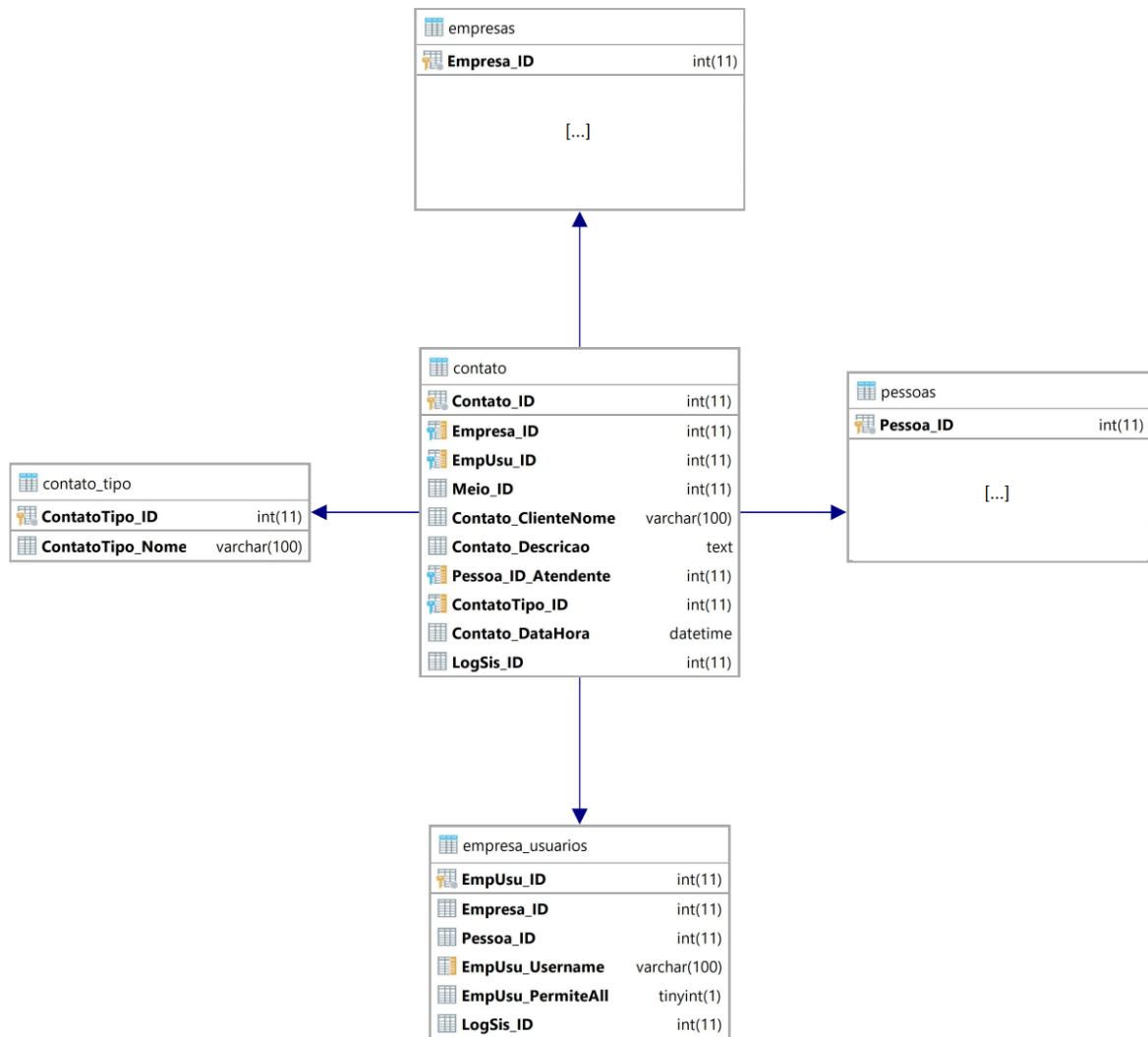


Figura 10 – Diagrama da tabela "contato" e suas tabelas relativas

Os campos "Ticket_Titulo", "Ticket_Descricao" e "Ticket_DataHora" são utilizados para registro do título, descrição e da data e da hora em que o ticket foi criado, respectivamente. Os campos "Pessoa_ID_Responsavel" e "Pessoa_ID_Atendente" são chaves estrangeiras da tabela "pessoas", referenciando respectivamente o desenvolvedor e o atendente responsáveis pelo ticket, conforme definido no capítulo 4. Os campos "TickTipo_ID", "TickStatus_ID", "TickPlataforma_ID" e "TickClass_ID" são chaves estrangeiras registrando a identificação das informações do tipo, status, plataforma e previsão de duração do ticket, respectivamente, criando uma ligação com tabelas auxiliares que listam cada uma das opções de cada campo, conforme também definido no capítulo 4. O Campo opcional Ticket_Deadline registra o *deadline* do ticket, caso algum tenha sido passado ao cliente requisitante. Os Campos "Ticket_Nivel", "Ticket_Previsao" e "Ticket_PrevisaoManual" são utilizados para registro do nível de prioridade do ticket, e de suas previsões de finalização,

funcionalidade que é descrita na seção 5.3.

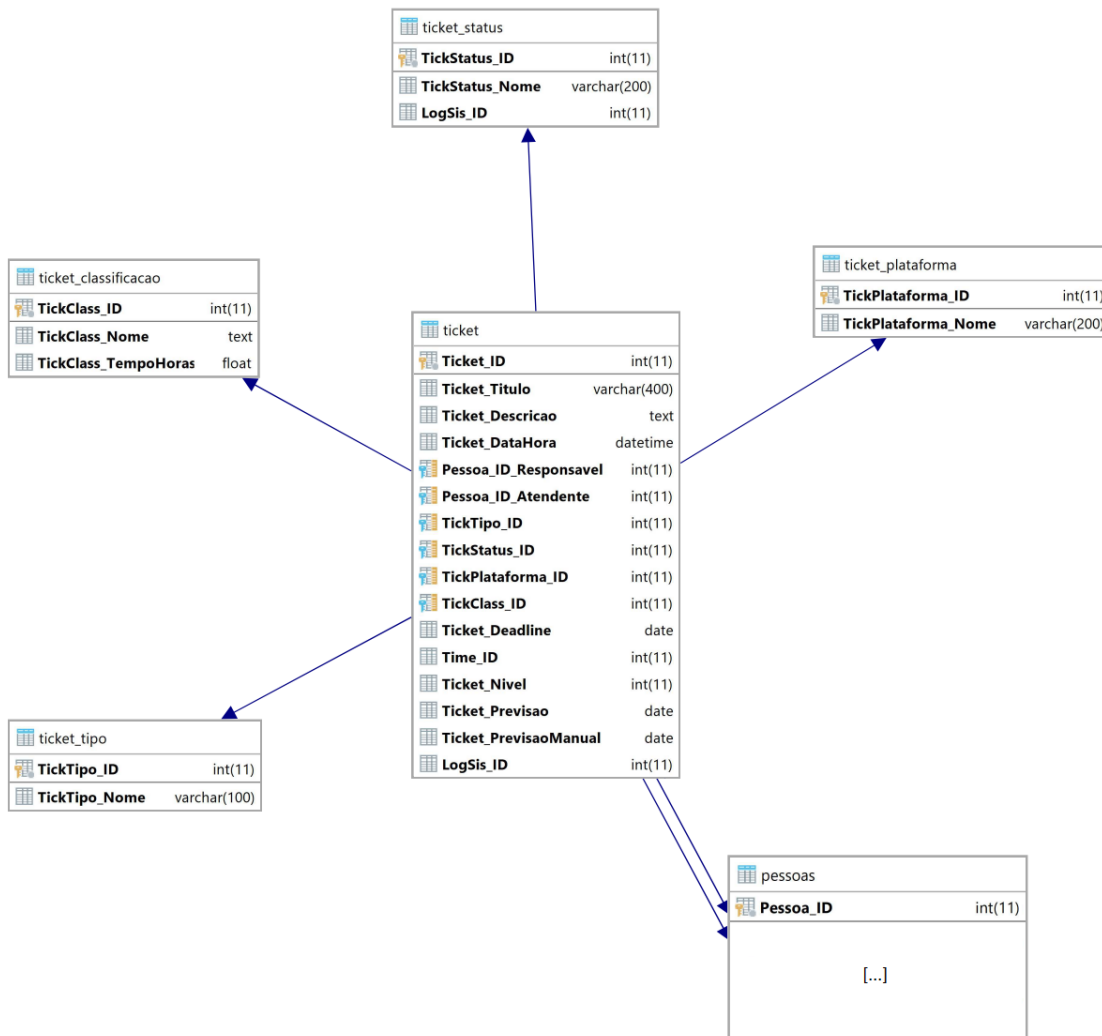


Figura 11 – Diagrama da tabela "ticket" e suas tabelas relativas

5.3 O Sistema de Estimativas

Após a implementação inicial da plataforma e seu uso por algumas semanas, verificou-se que o sistema de tickets funcionava bem. Porém, muitas *deadlines* passadas para clientes não eram cumpridas, principalmente devido à considerável quantidade de tickets sendo criados semanalmente, e ao fato de que as *deadlines* passadas eram estimadas de forma livre pelos desenvolvedores, sem nenhum dado de referência ou cálculo de prazos. Desta forma, julgou-se necessário implementar uma funcionalidade de estimativa de data de finalização para cada ticket registrado na plataforma, de forma que a própria plataforma pudesse gerar esta informação, com base nos tickets pendentes e em suas estimativas de duração.

Para que seja possível fazer uma estimativa de quando um ticket estará pronto, primeiramente precisa-se saber quando os trabalhos associados aquele ticket começarão. Assim, foi desenvolvido um sistema de filas de ticket, sob o conhecido princípio FIFO (*First In, First Out*) ou seja, o primeiro ticket a entrar na fila será o primeiro a sair. Porém, devido à existência de diferentes níveis de urgência dos tickets (alguns são sugestões de funcionalidades novas, sem urgência, e outros podem ser extremamente urgentes, como problemas na criação de uma matrícula), foram criadas três filas diferentes, uma para cada nível de prioridade, conforme descrito a seguir:

- Nível 1 - Prioridade máxima, tickets que devem ser feitos antes de todos os outros, contendo problemas críticos que impeçam o uso do sistema e afetam múltiplos clientes.
- Nível 2 - Prioridade Intermediária, tickets que devem ser resolvidos após os tickets do nível 1 acabarem. Para problemas ou questões que impeçam ou atrapalhem o uso normal do sistema por uma ou mais escolas.
- Nível 3 - Prioridade baixa, para modificações ou adição de funcionalidades não essenciais para escola, sugestões, entre outros. Os tickets desta fila só seriam lidados após o fim das outras duas filas.

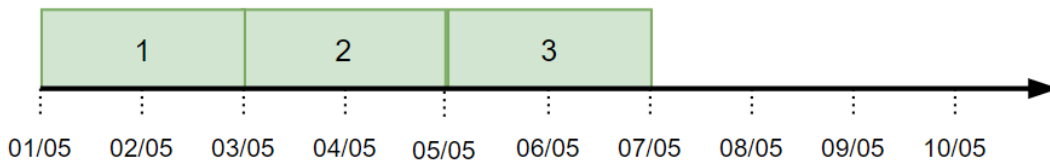
Dessa forma, o princípio FIFO é obedecido dentro de cada fila, mas um ticket novo de nível 1 entrará na frente de todos os tickets de nível 2 e 3.

Uma vez que os tickets estiverem organizados em filas, pode-se, a partir das estimativas de duração (tempo que o ticket levará para ser resolvido), estimar quando um ticket a entrar no final da fila poderá ser abordado, simplesmente somando a duração de cada ticket presente anteriormente na fila. Porém, foi observado que este cálculo só é válido quando não é possível a resolução de mais de um ticket ao mesmo tempo (ou seja, cada fila deve ter um desenvolvedor apenas). Devido a este fato, foi decidido organizar filas individuais, dos 3 níveis supracitados, para cada desenvolvedor, de forma que todo ticket criado é classificado em um dos três níveis e então delegado para um desenvolvedor específico, entrando no final da fila do respectivo nível daquele desenvolvedor.

Uma exceção ao sistema de filas existe quando há tickets com *deadline* passado para o cliente, que não seria cumprido se o ticket entrasse no final da fila ao ser registrado. Foi decidido pelos diretores da empresa que um ticket com *deadline* deve ter seu *deadline* cumprido, mesmo que isso implique que este ticket "fure" a fila existente, sendo abordado antes de outros que surgiram primeiro, porém não tem *deadline*. Assim, foi determinado que tickets com *deadline* "passariam na frente" de outros tickets da fila, porém apenas de um certo número de tickets suficiente para que o *deadline* seja cumprido. Caso, por exemplo, um novo ticket com *deadline* sendo inserido em terceiro lugar em uma dada fila

de 10 tickets seja suficiente para que sua previsão de finalização calculada seja anterior à sua *deadline*, não há necessidade de que ele entre em segundo ou primeiro lugar. Um diagrama contendo dois cenários simples desta organização de fila pode ser visualizado na figura 12

Cenário 1 - Tickets sem *deadline*



Linha do Tempo

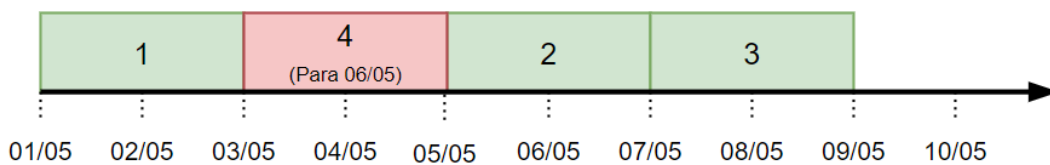
Previsões de Finalização:

Ticket 1 - dia 03/05

Ticket 2 - dia 05/05

Ticket 3 - dia 07/05

Cenário 2 - Ticket 4 com *deadline* para dia 06/05



Linha do Tempo

Previsões de Finalização:

Ticket 1 - dia 03/05

Ticket 4 - dia 05/05

Ticket 2 - dia 07/05

Ticket 3 - dia 09/05

Diagrama simplificado para facilitar visualização, considerando todos os tickets com duração estimada de 2 dias, e ignorando fins de semana.

Considerando dia 30/04 como data de finalização do ticket em andamento do desenvolvedor responsável por esta fila, pra que o próximo da fila comece a ser trabalhado no dia 01/05 (Ticket 1).

Figura 12 – Diagrama com cenários simples de organização de fila de tickets

Para lidar com este encaixe de filas e *deadlines*, foi desenvolvido um algoritmo de gestão de filas que leva em consideração as *deadlines* dos tickets, suas durações estimadas

e suas ordens de chegada, de acordo com as regras especificadas. Esta organização é feita dentro de cada nível de prioridade, nas filas pessoais de cada desenvolvedor, de forma automática sempre que um ticket é criado ou tenha seu nível de prioridade ou *deadline* alterados.

O Algoritmo de gestão destas filas funciona definindo para cada ticket uma previsão, de acordo com os parâmetros supracitados, e registrando isso no campo "Ticket_Previsao", citado na seção 5.2.2, para ser mostrado nas listas e tabelas da Plataforma, conforme mostrado nas próximas seções.

Uma vez que um ticket é marcado como em andamento, ele deixa de estar em qualquer uma das filas, e neste momento o desenvolvedor responsável é requisitado a definir uma data de finalização manual para o ticket, para que o próximo ticket da fila tenha uma data de quando ele deverá ser abordado. Dessa forma, todos os tickets da fila deste desenvolvedor terão sua previsão definida a partir da data em que ele estará livre (terminará o ticket que está em andamento), tendo assim, potencialmente, uma boa precisão.

Caso haja a necessidade de um desenvolvedor interromper o trabalho em um ticket e lidar com outro ticket mais urgente, o ticket interrompido pode ser colocado de volta em sua fila original (alterando seu status para "Em Aberto"), ou pode ser colocado na lista de "Em espera", alterando também seu status para tal. Os tickets classificados como "Em espera" não são considerados em nenhuma fila e em nenhum cálculo de prazos, devido à sua natureza incerta em termos de previsões.

5.4 Interface

5.4.1 A Página Escola

A visualização de uma escola (cliente da ValinSW) é feita na tela Escola, mostrada na figura 13. Através desta tela, todas as ações de registro de contato e criação de tickets podem ser efetuadas, de acordo com a especificação do software.

Para chegar nesta tela, mostrando uma uma escola específica, há dois modos:

- Pela busca de escolas, presente na extremidade superior direita da tela, conforme pode ser visto na figura 13. Ao digitar o nome de uma escola neste campo, e selecionar a escola, imediatamente a tela da escola é aberta, independente de qual aba está selecionada na barra superior.
- Pela tela inicial do sistema Manager, ao clicar na logo de qualquer escola presente nesta tela, a visualização da escola é aberta de imediato, possibilitando o rápido acesso a mesma.

Atendimento Começar Time Tracking Erros

Criados Hoje: 0 Resolvidos Hoje: 0 A Subir: 0

Resumo Pendências **Meus Tickets** Escolas Finalizados Estatísticas

Busca de Escolas

ESCOLA DE MÚSICA RAFAEL BASTOS Escola de Música Rafael Bastos

ID: 29
Florianópolis, Santa Catarina

Cadastro
Atendimento Antigo
Faturas

Contatos Tickets (2)

Registrar

WhatsApp

12 registros de 12 encontrados

Quem	O que	Tickets	Quando
Rafael WhatsApp	Leonardo Está com problemas ao encaminhar atendimentos da tela de novos atendimentos ao mauricio, usuario apenas com permissão de vendas.	✓ 1 ticket	25/10 às 16:31 Há 13 dias
Rafael WhatsApp	Leonardo Questiona sobre as aulas que estão com horário errado.	✓ 1 ticket	15/10 às 11:33 Há 23 dias
Rafael WhatsApp	Leonardo Deseja filtrar a disponibilidade por multiplas salas, para criar estatísticas referente a horários não ofertados aos clientes.	1 ticket	11/10 às 10:19 Há 27 dias
Rodrigo WhatsApp	Leonardo Aplicativo do iPhone está apresentando falhas	✓ 1 ticket	10/10 às 20:20 Há 28 dias
Rodrigo WhatsApp	Leonardo Enviou mensagem agradecendo, não especificou o quê.		08/10 às 15:55 Há 30 dias
Rafael WhatsApp	Leonardo Contato para relatar problemas com o trancamento de matrícula da aluna [REDACTED] e questionar como alterar a data inicial de um contrato.	✓ 1 ticket	02/10 às 10:19 Há 36 dias

Tickets Associados

Issue

Figura 13 – A Página Escola

5.4.1.1 Registro de Contato

O registro de um contato efetuado entre a empresa e um cliente (escola) é feito no campo presente no centro da tela. Ao se acessar esta tela, o foco da página é imediatamente colocado neste campo, de forma que pode-se começar a digitar de imediato. Após o fim da digitação do contato, descrevendo o que foi falado, deve-se selecionar qual dos funcionários registrados desta escola entrou em contato, no campo de seleção a direita, e então selecionar a forma de contato utilizada, entre WhatsApp, Ligação ou E-mail. Este campo já vem pré-selecionado com "WhatsApp", por ser o modo de contato mais utilizado pelos clientes da empresa atualmente. Uma vez selecionadas estas informações, pode-se clicar no botão "Registrar" e imediatamente o contato estará salvo na lista de contatos logo abaixo.

5.4.1.2 Lista de Contatos

A tabela de contatos que existe abaixo é a principal visualização da ValinSW sobre os últimos contatos realizados para com uma empresa cliente. Cada coluna desta tabela contém informações primordiais.

A primeira coluna contém o primeiro nome do cliente que realizou o contato, e a forma de contato utilizada para a comunicação.

A segunda coluna contém o primeiro nome do atendente, colaborador da empresa, que realizou o contato com o cliente especificado na primeira coluna. Além do nome, está também nesta coluna o registro que foi feito, descrevendo o que foi falado neste contato.

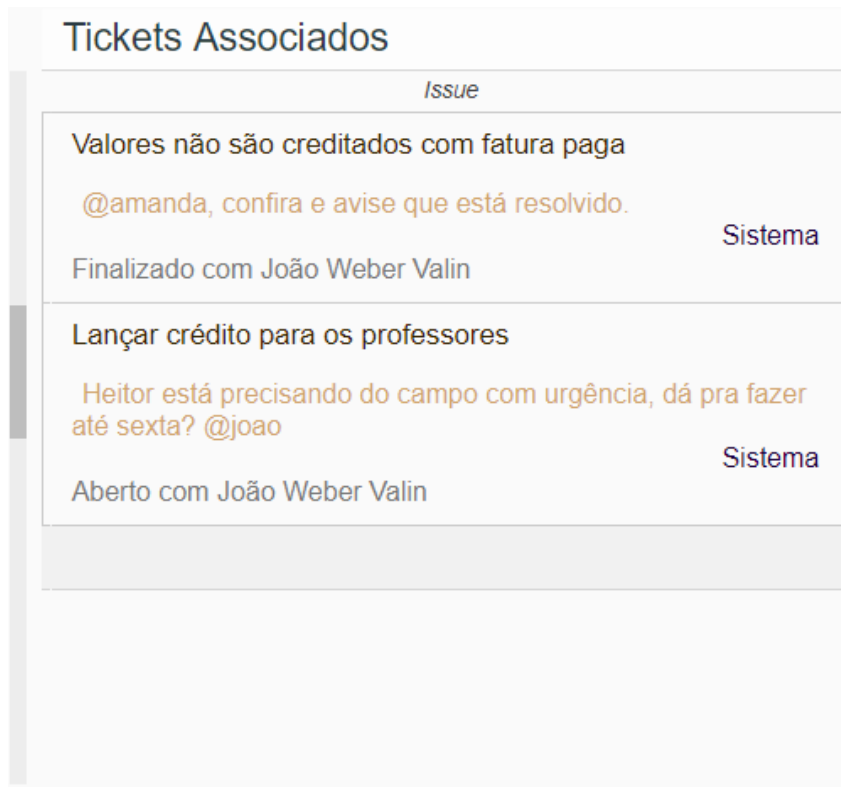
A terceira coluna, nomeada "Tickets", apresenta o número de tickets, ou seja, pendências, associadas àquele contato. Caso esta coluna esteja em branco, não houve nenhuma pendência criada a partir do contato, ou seja, este contato pode ter sido para tirar dúvida ou requisitar ajuda em alguma tarefa do sistema. Além do número de tickets associados, foi utilizado um código de cores e ícones para indicar a situação atual dos tickets ali indicados. Caso todos os tickets associados a este contato já estejam finalizados, o texto ficará verde e um ícone de "certo" aparecerá ao lado, conforme presente na figura 14. Caso nenhum tickets ali indicado já esteja resolvido, o texto ficará vermelho e nenhum ícone aparecerá. Caso haja mais de um ticket associado àquele contato, e nem todos estejam resolvidos mas também nem todos estejam pendentes, o texto ficará amarelo, sem nenhum ícone.

12 registros de 12 encontrados		1 / 1		30
Quem	O que	Tickets	Quando	
Rafael WhatsApp	Leonardo Está com problemas ao encaminhar atendimentos da tela de novos atendimentos ao mauricio, usuario apenas com permissão de vendas.	✓ 1 ticket	25/10 às 16:31 Há 13 dias	📄
Rafael WhatsApp	Leonardo Questiona sobre as aulas que estão com horário errado.	✓ 1 ticket	15/10 às 11:33 Há 23 dias	📄
Rafael WhatsApp	Leonardo Deseja filtrar a disponibilidade por multiplas salas, para criar estatísticas referente a horários não ofertados aos clientes.	1 ticket	11/10 às 10:19 Há 27 dias	📄
Rodrigo WhatsApp	Leonardo Aplicativo do iPhone está apresentando falhas	✓ 1 ticket	10/10 às 20:20 Há 28 dias	📄
Rodrigo WhatsApp	Leonardo Enviou mensagem agradecendo, não especificou o quê.		08/10 às 15:55 Há 30 dias	📄
Rafael WhatsApp	Leonardo Contato para relatar problemas com o trancamento de matrícula da aluna [REDACTED], e questionar como alterar a data inicial de um contrato.	✓ 1 ticket	02/10 às 10:19 Há 36 dias	📄

Figura 14 – Lista de Contatos da Página Escola

A terceira coluna contém também um link, de forma que, ao clicar nesta coluna em uma das linhas dos contatos, aparecerá, na pequena lista ao lado, os tickets associados a este contato, juntamente com algumas informações importantes sobre cada um, conforme listado a seguir. Um exemplo da lista de tickets associados pode ser visualizado na figura 15.

- Nome do Ticket;
- Plataforma do ticket (Sistema, Android, iPhone);
- A última anotação registrada naquele ticket, presente com o objetivo de mostrar facilmente o status atual do ticket, sem a necessidade de abrir a tela dele;
- O status e responsável pelo ticket.



Tickets Associados	
<i>Issue</i>	
Valores não são creditados com fatura paga <i>@amanda, confira e avise que está resolvido.</i>	Sistema
Finalizado com João Weber Valin	
Lançar crédito para os professores <i>Heitor está precisando do campo com urgência, dá pra fazer até sexta? @joao</i>	Sistema
Aberto com João Weber Valin	

Figura 15 – Lista de tickets associados a um contato

A partir da lista de contatos e desta visualização dos tickets associados a cada um em uma mesma tela, pode-se ter uma rápida visão geral sobre o status, tempo de espera e situação de cada contato e pendência que a ValinSW tem para com seus clientes, o que ajuda diretores e desenvolvedores a tomar decisões sobre pendências a serem abordadas antes de outras, e ajuda atendentes a ter certeza de que não há nenhuma pendência "esquecida" com cada escola.

A quarta coluna contém a data e a hora em que aquele contato foi registrado, juntamente com a descrição de há quantos dias isso ocorreu, para que a visualização e entendimento de há quanto tempo algo está pendente, por exemplo, seja mais fácil.

A quinta coluna contém apenas um ícone, para representar que existe ali um link que pode ser clicado. É através deste botão que se registra um ticket, conforme descrito na próxima seção deste documento.

5.4.1.3 Registro de Ticket

Uma vez registrado o contato, conforme descrito acima, o atendente pode então registrar um ticket associado a um contato. Este registro é feito clicando no ícone presente na quinta coluna da lista de contatos, que abre três opções, descritas a seguir:

- Associar a Ticket Existente - Caso aquele contato diga respeito a alguma requisição já feita anteriormente por outras escola ou pela mesma escola, onde há atualização de informações, assim pode-se associar o contato a um ticket já existente. Esta opção abre uma tela de busca pelos tickets existentes onde o atendente pode encontrar o ticket que gostaria de associar e associá-lo com um clique. Esta tela não será mostrada aqui por simplicidade.
- Editar Contato - Onde pode-se editar o contato representado pela linha na tabela onde o ícone foi clicado. Esta opção abre uma pequena tela modal que permite alterar informações do contato como: A forma de contato utilizada, o nome do cliente que entrou em contato ou o conteúdo do contato, por razão de algum erro ou mesmo da necessidade de atualização de um contato existente.
- Criar novo Ticket - Esta opção, ao ser selecionada, abre uma tela modal através da qual pode-se registrar um ticket, conforme pode ser visto na figura 16, e é descrita a seguir.

Apesar do foco constante no desenvolvimento da aplicação em tornar toda operação realizada rápida e prática, a criação do ticket, devido às informações necessárias, é a parte mais demorada. Ainda assim, não leva-se mais do que 30 segundos para registrar um ticket novo.

No formulário de registro de ticket, conforme mostrado na figura 16, há o nome do cliente e a escola requisitante deste ticket, e esta informação já vem preenchida. As outras informações, descritas a seguir, devem ser preenchidas.

O Resumo do ticket, ou seja, seu título, utilizado para listagem de tickets e fácil identificação.

A Descrição do ticket, com o máximo de informações possível para auxiliar o desenvolvedor que posteriormente lidará com o ticket. Este campo vem já pré-preenchido com a descrição do contato realizado, mas o atendente poderá modificar, adicionando detalhes específicos ou retirando informações não pertinentes a este registro.

O tipo do ticket, podendo ser "Problema", "Modificação / Algo Novo", "Intervenção", "Sugestão" ou "Outro".

A plataforma do ticket, podendo ser "Sistema", "Android" ou "iPhone". Esta informação é importante pois direcionará o ticket à fila do responsável geral de cada plataforma, conforme explicado na subseção 5.3

O Nível de prioridade do ticket, podendo ser do nível 1 ao nível 3, e cada seleção de nível mostra sua descrição. A seleção do nível de prioridade do ticket é utilizada para a estimativa de sua previsão de finalização, que será passada para o cliente caso ele peça.

Novo Ticket

Cliente: Rafael da Escola de Música Rafael Bastos

Resumo:

Questiona sobre as aulas que estão com horário errado.

Tipo: Problema

Plataforma: Sistema

Nível de Prioridade: Nível 1 Nível 2 Nível 3

Prioridade Intermediária
Ticket é sobre algo que afeta ou pode afetar uma ou algumas escolas. Algo que atrapalhe a utilizar o sistema normalmente ou atrapalhe o modo com que a escola funciona

Previsão: **quarta, 26 de dezembro de 2018**

Foi passado deadline para o cliente Sim Não

Deadline: 26/12/2018

Figura 16 – Janela de registro de um novo Ticket, a partir de um contato

Finalmente, há a definição se foi passado um *deadline* para o cliente, sobre este ticket, ou não. Não são todos os clientes que pedem uma estimativa de prazo, mas caso seja solicitada e informada de acordo com a previsão dada, esta *deadline* deve ser registrada para posterior organização e priorização de ticket para cumprir o *deadline* passado.

Após a especificação destas informações, ao clicar no botão "Confirmar", o ticket é registrado e o sistema é redirecionado para a visualização completa deste ticket, descrita na seção a seguir.

5.4.2 A Página Ticket

A página Ticket tem o propósito de conter todas as informações pertinentes a um ticket, e ser o meio pelo qual pode-se modificar, atualizar e registrar informações do ticket. Uma imagem desta tela pode ser vista na figura 17.

A primeira informação presente na tela de ticket é o título e a descrição deste ticket,

em destaque na parte superior da tela. Abaixo desta informação, no canto direito, está presente um botão para edição do título e a descrição do ticket, caso necessário. No canto esquerdo, temos a barra de botões de ações rápidas, cujo conteúdo é alterado de acordo com a situação atual do ticket. Esta funcionalidade será descrita mais adiante.

The screenshot displays the 'Ticket' management interface. At the top, there's a header with 'Ticket', 'Desenvolvimento', and 'Informações Gerais'. Below the header, the interface is divided into two main sections. The left section features a text area with the heading 'Incluir Transferência em tipos de pagamento' and a description 'Pediu que criasse a forma de pagamento preferível transferência.' Below this is an 'Adicionar Anotação' button and a list of activity logs. The right section is a form for 'Informações Gerais' with fields for Platform (Sistema), Type (Modificação / Algo Novo), Status (Finalizado), Estimated Time (< 2 Hora), Level (Nível 1 - Problema Geral), Developer (João Valin), Attendant, Client Deadline (00/00/0000), and Prediction (04/12/2018). At the bottom of the right section, there's a field for the Requirer (Casa Talento 25 (Stephany)).

Figura 17 – A Página Ticket

5.4.2.1 A Coluna de Informações Gerais

A coluna de informações gerais, presente no lado direito da tela de ticket, tem como objetivo informar e possibilitar a fácil alteração de todas as classificações e informações pontuais do ticket, conforme definido nas especificações da plataforma. Ali, pode-se visualizar e alterar a plataforma do ticket, seu tipo, status, tempo estimado e nível. Ali, pode-se alterar também o desenvolvedor responsável por aquele ticket, e o atendente responsável por entrar em contato com o cliente requisitante, caso necessário posteriormente. Abaixo dessas opções, há também um campo para visualização e alteração do *deadline* passado para o cliente, caso este tenha que ser alterado e passado um novo para o cliente, e a visualização da previsão de finalização do ticket, gerada automaticamente conforme explicado na seção sobre o sistema de filas.

Por fim, há uma lista de contatos relacionados a este ticket, contendo os nomes das escolas e dos clientes requisitantes. Ou seja, pode-se visualizar todas as escolas que estão aguardando a resolução daquele ticket.

5.4.2.2 Botões de Ação Rápida

Os botões de ação rápida da página Ticket tem como objetivo facilitar e acelerar o uso da plataforma por desenvolvedores e atendentes fornecendo as ações mais utilizadas para cada status em que o ticket se encontra. As situações e qual botão é colocado nesta parte estão descritos a seguir:

- Ticket com outro responsável: O botão "Delegar pra mim", o qual automaticamente delega o ticket para o usuário atual, mantendo seu status;
- Ticket em aberto delegado para o usuário: Botão "Começar", o qual muda o status do ticket para "Em Andamento";
- Ticket em andamento delegado para o usuário: Botão "Pode Subir", o qual muda o status do ticket para "Aguardando Deploy", um status intermediário antes da finalização;
- Ticket aguardando deploy delegado para o usuário: Botão "Finalizar", o qual muda o status do ticket para "Finalizado".

5.4.2.3 Registro de Anotações e Menções

Na parte central da tela de ticket, há um campo onde pode-se inserir anotações diversas ao ticket, e então clicar no botão "Adicionar Anotação" para registrá-la. Este botão é duplo, e sua segunda metade, de cor laranja, pode ser utilizada para, além de registrar a anotação, passar a responsabilidade do ticket para o atendente. Esta opção é chamada "passar para o atendente", e é utilizada principalmente para quando um desenvolvedor que está trabalhando em uma certa funcionalidade ou modificação precisa de uma informação do cliente. Assim, o desenvolvedor pode escrever uma anotação explicando a informação que ele necessita, e então utilizar a opção supracitada para notificar o atendente responsável.

O campo de registro de anotações também contém a funcionalidade de mencionar colegas de trabalho, para comunicação interna. Para mencionar alguém, basta incluir, no corpo da anotação, o *username* do colega, precedido do caractere "@". Isto fará com que o usuário mencionado receba uma notificação, e poderá responder a anotação com outra anotação posteriormente. Um exemplo de uso real desta funcionalidade pode ser visto na figura 18.

Abaixo do campo de registro de anotações, há a lista de anotações, registros de histórico e contatos realizados relacionados aquele ticket. É uma tabela contendo duas colunas, uma com o conteúdo, conforme indicado abaixo, e outra com a data e hora em que o registro aconteceu. A tabela é ordenada pela data hora em que o registro aconteceu, de forma decrescente, e contém 3 tipos de registros:

- Registro de alteração de alguma propriedade do Ticket - Em caso de alguma propriedade do ticket (presentes na coluna de informações gerais) ser alterada, é criado um registro no histórico do ticket, e que portanto aparecerá nesta tabela contendo uma descrição de qual propriedade foi alterada, pra qual valor e por quem.
- Contatos associados a este ticket - Todos os contatos realizados com clientes que estejam associados ao ticket também aparecerão nesta lista, em ordem cronológica, e devidamente indicados como contato, incluindo o nome do cliente e da escola a qual ele pertence. Assim, desenvolvedores que estejam trabalhando neste ticket estarão cientes de novos contatos realizados sobre aquela questão, ao longo do ciclo de vida do ticket.
- Anotações comuns - Anotações realizadas pelos colaboradores da empresa, incluindo a anotação e o primeiro nome de quem a realizou.

Um exemplo desta tabela contendo todos os tipos de registros listados acima pode ser visto na figura 18



<input type="text"/> Adicionar Anotação	
Leonardo Finalizada a geração, enviado para homologação. Aguardando resposta.	02/10 11:09
João alterou a deadline para 05/10/2018	29/09 13:32
João alterou a deadline para 26/10/2018	29/09 13:32
Leonardo @matheus homologação da caixa tá parada, mas eu cuido disso depois do Sicredi e Ailos	26/09 12:57
Leonardo delegou para Leonardo	26/09 12:56
Matheus delegou para	25/09 19:55
Matheus alterou o tempo estimado para Semanas	25/09 19:55
Matheus @greg @joao alguem ta cuidando disso? como esta?	25/09 19:55
Ticket criado dia 24/07 10:08 por João	24/07 10:08
Contato de Lucicleide da Conservatorio de Musica Sonata: Integração de boletos com a caixa.	24/07 10:08

Figura 18 – Exemplo de anotações, históricos e contatos registrados em um ticket

5.4.3 A Página Pendências

A tela de pendências foi desenvolvida com o proposito de agrupar todas as pendências (tickets) que a empresa tem para serem resolvidas. Estes tickets foram agrupados em tabelas, contendo todas as informações pertinentes para visualização rápida e compreensão

da situação de cada um. Os tickets nesta visualização são agrupados em 4 categorias: uma para cada nível de prioridade, conforme definido anteriormente, do nível 1 ao 3, e aos tickets já iniciados (em andamento, em espera ou aguardando deploy). A tela de pendências pode ser visualizada na figura 19.

Previsão	Ticket	Tipo	Criado há	Desenvolvedor	Plataforma	Quem
5 dias	Criação de relatório de desenvolvimento <small>Conversei com o cliente, os itens que aparecem no relatório são classificados de A até D, e as avaliações são feitas pelo próprio professor, portanto os campos dependem apenas do professor definir.</small>	Modificação / Algo Novo	37 dias	Leonardo	Sistema	IN CONCERT ESCOLA DE MÚSICA
6 dias -3	Turmas com instrumentos diferentes	Modificação / Algo Novo	33 dias	Matheus	Sistema	Viver de Música
7 dias 0	Parcelamento de débito do professor <small>adicionar meses pra lançamento futuro</small>	Modificação / Algo Novo	64 dias	João	Sistema	GTR Instituto de Música Asa Sul
7 dias	Incluir reagendamentos no relatório	Modificação / Algo Novo	36 dias	Leonardo	Sistema	Espaço Musical Vila Lobos
8 dias	Adicionar Eventos na agenda	Modificação / Algo Novo	85 dias	Matheus	Sistema	Escola de Música NVA Espaço Musical Vila Lobos Conservatório de Música Mozart
9 dias -1	Não permitir reagendo do aluno antes de 48 horas . <small>@greg verifica pra nós oq exatamente esse cara quer nos dois tickets desta escola (são parecidos)</small>	Modificação / Algo Novo	76 dias	João	Sistema	Escola de Música Straus
12 dias	O Pedagógico somente anotação em atendimento do aluno	Modificação / Algo Novo	79 dias	Matheus	Sistema	Escola de Música NVA
13 dias	Diferenciar recessos e feriados					

Figura 19 – A Página Pendências

Além do agrupamento pelo status do ticket, estes também podem ser filtrados pela plataforma e/ou pelo desenvolvedor responsável, através dos botões presentes no canto superior direito da tela.

5.4.3.1 A Tabela de Tickets Pendentes

A tabela de tickets pendentes presente na tela de pendências é densa e contém muitas informações.

Na primeira coluna, chamada de "previsão", há o número de dias restantes até a previsão da finalização do ticket. Esta previsão é calculada usando o algoritmo de filas apresentado anteriormente. Já a cor em que esta informação é apresentada depende da situação da previsão com relação ao deadline do ticket. Caso o ticket não possua um deadline registrado (não foi passado ao cliente), a previsão é dada sempre na cor verde. Caso o ticket tenha um deadline, a cor depende da situação: Caso a previsão seja antes do deadline, a informação é apresentada em verde. Caso a previsão seja no mesmo dia do deadline, a informação é apresentada em amarelo. Caso a previsão seja após o deadline, a informação é apresentada em vermelho, indicando que o ticket está potencialmente

atrasado. Além disso, caso haja o deadline, um pequeno número é mostrado no canto desta célula da tabela indicando a diferença, em dias, entre a previsão e o deadline, conforme destacado na figura 20.

Previsão	Ticket
5 dias	Criação de relatório de desenvolvimento Conversei com o cliente, os itens que aparecem no relatório até D, e as avaliações são feitas pelo próprio professor, dependem apenas do professor definir.
6 dias -3	Turmas com instrumentos diferentes
7 dias 0	Parcelamento de débito do professor adicionar meses pra lançamento futuro
7 dias	Incluir reagendamentos no relatório

Figura 20 – Destaque da lista de Pendências

Na segunda coluna da tabela, há o título do ticket e a última anotação registrada neste ticket, caso haja alguma. Ao clicar nesta coluna em uma linha da tabela, o usuário é levado à tela Ticket, descrita anteriormente, daquele ticket em questão.

As demais colunas da tabela informam informações pontuais sobre o ticket, na seguinte ordem: O tipo do ticket, há quantos dias ele foi registrado, quem é o desenvolvedor responsável por aquele ticket, a plataforma do ticket e, por fim, as escolas que estão aguardando a resolução daquele ticket. Ao clicar na última coluna, o usuário é levado à tela Escola da primeira escola que requisitou aquele ticket.

5.4.4 A Página de Tickets Delegados

A aba de tickets delegados, chamada de "Meus Tickets", foi criada para centralizar as operações para usuários desenvolvedores. Nesta tela pode-se visualizar todos os tickets que estão delegados para um dado usuário, organizadamente de acordo com seu status ou nível de prioridade. A tela de "Meus Tickets" é separada em três seções, que podem ser alternadas utilizando o componente de seleção, conforme indicado na figura 21

5.4.4.1 Em Andamento

The screenshot shows a dashboard titled 'Atendimento' with a search bar and navigation buttons like 'Começar', 'Time Tracking', and 'Erros'. It displays statistics for 'Criados Hoje: 0', 'Resolvidos Hoje: 0', and 'A Subir: 0'. The main section is 'Meus Tickets' with tabs for 'Resumo', 'Pendências', 'Escolas', 'Finalizados', and 'Estatísticas'. Below are radio buttons for 'Dash', 'Fila', and 'Notificações', and a dropdown for 'Sob responsabilidade de Leonardo Ronconi'. The dashboard is divided into three columns: 'Em Andamento', 'Em Espera', and 'Aguardando'.

Em Andamento		Em Espera		Aguardando
Issue	Data de Finalização	Issue	Data de Finalização	
Alterar Bandas para Reuniões	2 dias	Boleto Sicoob	Amanhã	
Backup do sistema	Hoje	Boleto com Protesto	9 dias	
		Integração de boletos com a caixa. <small>Recobido email com respostas da homologação. já feitas as correções e enviado</small>	Amanhã	
		Boletos Banco do Brasil <small>Corrigido os valores para o arquivo retorno, aguardando darlan confirmar a sequencia do ultimo boleto.</small>	-12 dias	

Figura 21 – A Página "Meus Ticket- Seção Painel

Na seção em andamento, o usuário pode visualizar todos os tickets já iniciados sob sua responsabilidade, separados por status em três pequenas listas, uma para os tickets em andamento, outra para os tickets em espera, e por ultimo uma para os tickets aguardando deploy. Estas listas são muito largas para caberem todas em uma mesma tela ao mesmo tempo, por isso existe a possibilidade de mover a tela horizontalmente para ver as outras listas.

Em cada lista, além do nome do ticket e da ultima anotação registrada nele, há também a data de finalização configurada pelo desenvolvedor.

Além das listas de tickets, há também uma pequena lista de tickets aguardando ação do usuário, que só aparece quando esta não está vazia. Toda vez que um usuário é marcado na anotação de um ticket, conforme explicado anteriormente, este ticket aparecerá nesta lista de "tickets aguardando minha ação". Esta funcionalidade foi desenvolvida devido a necessidade, muitas vezes, de um tempo para que o usuário possa responder a marcação. Muitas vezes, o usuário acabava esquecendo que ele havia sido marcado e não respondia, ou tinha dificuldade de encontrar o ticket no qual ele havia sido marcado. Dessa forma, toda marcação ficará registrada nesta lista, até que o usuário retire o ticket da lista. Isso pode ser feito através do botão presente em cada item da lista, na segunda coluna.

Issue	Previsão / Deadline	Tipo	Delegar	Mudar de Nível	Quem
Nível 1					
Retirar valores zerados do demonstrativo	2 dias	Modificação / Algo Novo	Delegar	Nível	GTR Instituto de Música Asa Sul
Nível 2					
Criação de relatório de desenvolvimento <small>Conversei com o cliente, os itens que aparecem no relatório são classificados de A até D, e as avaliações são feitas pelo próprio professor, portanto os campos dependem apenas do professor definir.</small>	6 dias	Modificação / Algo Novo	Delegar	Nível	IN CONCERT ESCOLA DE MÚSICA
Incluir reagendamentos no relatório	8 dias	Modificação / Algo Novo	Delegar	Nível	Espaço Musical Vila Lobos
Nível 3					
Criar aba "Crédito de Horas" para professores <small>É só colocar a aba q tem em pessoas na parte de professor</small>	9 dias	Modificação / Algo Novo	Delegar	Nível	ABC ESCOLA DE MÚSICA
Notificar alteração e cancelamento de experimental	9 dias	Modificação / Algo Novo	Delegar	Nível	GTR Instituto de Música Asa Sul

Figura 22 – A Página "Meus Ticket- Seção Fila"

5.4.4.2 Fila Pessoal

Na seção "fila", que pode ser visualizada na figura 22, há uma única tabela contendo todos os tickets pendentes (ou seja, com o status em "em aberto") sob responsabilidade do usuário. Os tickets são ordenados pelo nível, do nível 1 ao nível 3, e entre cada um dos níveis há um separador indicado a qual nível aqueles tickets pertencem. Dentro de cada seção de nível, os tickets são ordenados por sua previsão de finalização, de modo que esta finalização se cumprirá caso cada ticket seja desenvolvido na ordem, de cima pra baixo, no prazo esperado.

Na tabela, há as informações padrões de outras tabelas citadas anteriormente, como o título do ticket, sua última anotação, a previsão e sua relação com o deadline do ticket, o tipo do ticket e qual escola ou escolas estão aguardando esta resolução. Além disso, há botões para rapidamente alterar o responsável pelo ticket, ou seja, para delegá-lo para outra pessoa, e para mudar o nível de prioridade daquele ticket, conforme necessidade.

5.4.4.3 Notificações

Por fim, há a seção de notificações, contendo um histórico das últimas notificações enviadas para o usuário, incluindo tanto citações como delegações de ticket. Este histórico é uma tabela contendo, para cada notificação, as informações de quando ela foi enviada, quem enviou, sobre qual ticket era e qual a notificação. Esta seção pode ser visualizada na figura 23.

The screenshot shows a web interface for a ticketing system. At the top, there's a navigation bar with 'Atendimento' and a search bar. Below it, there are statistics: 'Criados Hoje: 0', 'Resolvidos Hoje: 0', and 'A Subir: 0'. There are also buttons for 'Começar', 'Time Tracking', and 'Erros'. The main content area has tabs for 'Resumo', 'Pendências', 'Meus Tickets', 'Escolas', 'Finalizados', and 'Estatísticas'. Under 'Meus Tickets', there are sub-tabs for 'Dash', 'Fila', and 'Notificações'. The 'Notificações' tab is active, displaying a table titled 'Histórico de Notificações'.

Quando	Quem mandou	Ticket	Deadline	Categoria	Status
Há 16 dias, 14:52	Amanda	Não está conseguindo alterar o curso, impedido de alterar os professores também. @matheus, Ainda não.		Sistema	Finalizado
Há 16 dias, 11:12	Amanda	Não está conseguindo alterar o curso, impedido de alterar os professores também. @matheus, Quando um aluno muda de instrumento, não está conseguindo alterar. quando clica na seta para descer as opções, mas fica travado no instrumento q já está cadastrado.		Sistema	Finalizado
Há 16 dias, 11:03	Amanda	Não está conseguindo alterar o curso, impedido de alterar os professores também. Ticket delegado pra você		Sistema	Finalizado
Há 21 dias, 10:00	João	Permitir reagendar aulas justificadas. Ainda não @valin	-23 dias	Sistema	Finalizado
Há 22 dias, 18:14	Matheus	Permitir reagendar aulas justificadas. @valin ta funcionando nos apps	-23 dias	Sistema	Finalizado
Há 22 dias, 13:39	Leonardo	Habilitar pdf no aplicativo @matheus pdf? pdf	2 dias	Sistema	Finalizado
Há 23 dias, 14:41	Leonardo	Turmas com instrumentos diferentes Ticket delegado pra você	3 dias	Sistema	Aberto
Há 23 dias,		Alunos de turma fora da turma		Sistema	Finalizado

Figura 23 – A Página "Meus Tickets- Seção Notificações

6 Análise dos Resultados

Uma vez que a primeira versão básica da plataforma foi finalizada, foram realizados diversos testes para garantir seu bom funcionamento. Estes testes foram principalmente de simulação de uso, incluindo a criação de contatos e tickets fictícios, realização de todas as ações previstas para tickets e contatos, simulando ciclos de vida de tickets e edições de contatos realizados.

A implementação da primeira versão da plataforma se deu de forma rápida, após um pequeno treinamento apresentado pelo aluno para os colaboradores da empresa na utilização da plataforma. Rapidamente os colaboradores passaram a utilizar a plataforma sem grandes problemas. A plataforma foi considerada, em geral, de fácil e rápida utilização, com tempos de espera extremamente pequenos e ações corriqueiras facilmente realizáveis. A partir desse ponto, diversas observações e sugestões foram passadas ao aluno pelos colaboradores, as quais foram gradualmente implementadas na plataforma durante um período de 3 semanas, melhorando de forma considerável a usabilidade geral da plataforma.

O Atendimento às escolas teve uma considerável melhora, com o registro de todos os contatos e tickets associados a uma escola facilmente acessível, atendentes podem ter uma visão geral de como está o relacionamento da empresa com esse dado cliente sempre que há um contato. Um problema específico resolvido com a implementação da plataforma foi o caso comum de um cliente ligar perguntando algo como "Como está a situação daquilo que conversamos semana passada?". Antes da implementação da plataforma, perguntas como essa eram um problema, uma vez que não havia registro dessa conversa, não se sabia com quem o cliente falou, e muitas vezes a pessoa com quem ele falou também não lembrava o que exatamente foi conversado. Com a plataforma, pode-se facilmente visualizar na lista de contatos da escola deste cliente, o que foi falado, se houve alguma pendência criada com relação àquele contato, qual o status atual desta pendência, qual o desenvolvedor responsável por ela e qual a data prevista de finalização da mesma.

O sistema de previsões de datas de finalização de tickets, apesar de nem sempre acertar, devido às incertezas intrínsecas do desenvolvimento de software com relação ao tempo necessário para realizar uma dada tarefa, funcionou bem e tem ajudado a melhorar o atendimento aos clientes. Havia, antes da implementação dessa funcionalidade, uma arbitrariedade na escolha de qual ticket abordar em seguida, para os desenvolvedores, o que causava uma incerteza quanto a quando algo seria resolvido, além de muitas vezes certos tickets mais complicados acabarem sendo evitados pelos desenvolvedores, fazendo com que demorassem muito tempo para serem resolvidos, com tickets novos surgindo todos os dias. Com o sistema de filas e previsões, há sempre indicado qual deve ser o próximo

ticket a ser abordado para que as entregas sejam mais rápidas e os *deadlines*, cumpridos.

Uma consequência positiva observada do algoritmo de filas, conforme descrito no capítulo 5, é a reorganização automática da ordem de tickets na fila em caso de atrasos na finalização de um dado ticket. Para que a *deadline* de um dado ticket na fila seja cumprido, há casos em que o algoritmo gradualmente (com o passar dos dias e o aumento do atraso do ticket em andamento) passe o ticket em questão para frente na fila, pois o tempo necessário para realização do ticket que estava antes deste já não existe, fazendo com que o ticket com *deadline* deva ser realizado primeiro para que a entrega seja de acordo com o *deadline*.

O sistema de registro de anotações com menções de colegas nas anotações de tickets também se provou muito útil, sendo utilizado grandemente e registrando até mesmo pequenas conversas sobre o desenvolvimento de tickets, associadas ao ticket em questão. Tal sistema de comunicação interno tem tornado o desenvolvimento dentro da empresa mais dinâmico e mais documentado, uma vez que pode-se consultar posteriormente o que foi falado e quais foram as anotações registradas em cada ticket.

Assim, a plataforma tem sido uma grande ajuda para a empresa desde que foi implementada. Até o momento de escrita deste documento, em menos de 2 meses, foram registrados 357 contatos entre a empresa e seus clientes, e 274 pendências (tickets) a partir destes contatos.

7 Conclusões e Perspectivas

Através do processo de desenvolvimento e implementação da plataforma proposta neste documento, foi possível ao aluno perceber que o desenvolvimento do início ao fim de um software a ser utilizado como ferramenta de trabalho no dia a dia das pessoas é algo mais complicado do que pode parecer. Ajustar a plataforma ao que as pessoas esperam, e ao processo de trabalho que elas já estão acostumadas é algo delicado de se fazer.

Apesar disso, com a boa vontade e a disposição dos colaboradores da empresa a ajudar no desenvolvimento desta plataforma, esta, em sua versão final (até o momento da escrita deste documento) foi bastante aceita e está sendo utilizada grandemente por todos os colaboradores da ValinSW.

Quanto a parte técnica, diversas disciplinas do curso de Engenharia de Controle e Automação foram decisivas para o sucesso no desenvolvimento deste projeto, como "Introdução à Informática para Automação", "Fundamentos da Estrutura da Informação", "Informática Industrial I", "Fundamentos de Sistemas de Banco de Dados" e "Metodologia para Desenvolvimento de Sistemas".

Havia, é claro, sempre a possibilidade da empresa ter adotado um software pronto para auxiliar nos processos de atendimento e suporte a cliente, pois existem muitos no mercado. Porém, a decisão de criar o próprio sistema, modelado especialmente para a empresa, tem sido considerada, agora que o sistema já está pronto e sendo utilizado, uma boa escolha. Conforme a empresa cresça e seus processos mudem, a plataforma poderá ser constantemente modificada e melhorada para sempre ser a melhor possível para as necessidades particulares da ValinSW, objetivo este que é o mesmo que a empresa tem com seus clientes, utilizando seu produto.

Por fim, a perspectiva com relação a esta plataforma é que ela cresça, modernize e amadureça conforme a empresa cresce e amadurece, sempre estando perfeitamente ajustada às necessidades e processos da empresa.

Referências

- 1 SOMMERVILLE, I. *Engenharia de Software*. 9. ed. [S.l.]: Pearson, 2011. Citado 5 vezes nas páginas 9, 23, 24, 29 e 30.
- 2 PHP: Hypertext Preprocessor. 2018. <<http://php.net>>. Acessado em 04/11/2018. Citado na página 23.
- 3 USAGE statistics and market share of PHP for websites. 2018. <<https://w3techs.com/technologies/details/pl-php/all/all>>. Acessado em 04/11/2018. Citado na página 23.
- 4 MYSQL. 2018. <<https://www.mysql.com/>>. Acessado em 04/11/2018. Citado na página 24.
- 5 COMPANIES Using MySQL. 2018. <<https://idatalabs.com/tech/products/mysql>>. Acessado em 05/11/2018. Citado na página 24.
- 6 W3C HTML. 2018. <<https://www.w3.org/html/>>. Acessado em 05/11/2018. Citado na página 24.
- 7 JAVASCRIPT. 2018. <<https://www.javascript.com/>>. Acessado em 05/11/2018. Citado na página 24.
- 8 JSON. 2018. <<https://www.json.org/>>. Acessado em 15/11/2018. Citado na página 24.