

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

GLAUBER HERMANY

**OnCall Manager – Uma aplicação para gerência de equipes de suporte a
produção de software**

Trabalho de Graduação.

Orientador: Prof. Dr. Marcelo Soares Pimenta

Porto Alegre
2015

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Graduação: Prof. Sérgio Roberto Kieling Franco

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do Curso de Ciência da Computação: Prof. Raul Fernando Weber

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Agradeço aos meus pais, meus tios e meus irmãos pelo apoio. Agradeço ao professor Marcelo Pimenta pela orientação no trabalho de graduação.

RESUMO

Este trabalho descreve o desenvolvimento de uma aplicação web para gerenciar equipes de suporte a produção de software. O objetivo da aplicação é informatizar alguns processos manuais da equipe de suporte aumentando a produtividade do time e a correteude dos resultados antes manuais. O processo de desenvolvimento do software passa pela análise dos requisitos, a definição da metodologia de desenvolvimento, a definição das tecnologias utilizadas no desenvolvimento do produto, a codificação do projeto e os testes sobre o código fonte do produto. No texto do documento são apresentadas as formas como cada uma dessas etapas contribuíram para transformar a ideia do software em um produto disponível para os usuários.

Palavras-chave: Suporte. Processo de desenvolvimento de software. Software de gerência.

OnCall Manager – an application to manage software production support teams

ABSTRACT

This work describes the development process to build a web application responsible to help a software support team organization. The main goal of this application called OnCall Manager is to automate some manual process from the support team. The automation will increase the team productivity and the internal process correctness. The development process includes the application requirements analysis, the development methodology definition, the definition of the technologies used in the product development, the product coding process and the tests over the the product source code. The document shows how each part of the development process was done to create the OnCall Manager software.

Keywords: Support. Software development process. Management software.

LISTA DE FIGURAS

Figura 3.1 – Arquitetura MVC do JavaServer Faces.....	14
Figura 3.2 – Parte do código do persistence.xml.....	15
Figura 3.3 – Relatório de cobertura de testes gerado pela ferramenta EclEmma.....	17
Figura 4.1 – Fluxo de execução do framework scrum.....	22
Figura 4.2 – Estórias no backlog do produto e lista de estórias aceitas.....	23
Figura 5.1 – web.xml do projeto com a configuração de filtros para autenticação de acesso..	26
Figura 5.2 – Diagrama de classes para o controle de acesso.....	27
Figura 5.3 – Tela de geração da próxima escala de plantão.....	28
Figura 5.4 – Diagrama de classes para o controle e geração de escalas de plantão.....	28
Figura 5.5 – Diagrama de classes para o controle de acionamentos.....	29
Figura 5.6 – Diagrama de classes para o controle da knowledge base.....	30
Figura 5.7 – Tela com requisitos necessários para um usuário estar apto a ser plantonista.....	31
Figura 5.8 – Diagrama de classes para Controle do checklist.....	31
Figura 5.9 – Planilha parcial com a horas de sobre aviso do período Abril-Maio 2015.....	33
Figura 5.10 – Diagrama de classe para geração da planilha financeira.....	34

LISTA DE TABELAS

Tabela 3.1 – Tabela comparativa entre funcionalidades das aplicações.....	19
Tabela 4.1 – Estórias implementadas em cada sprint.....	24
Tabela 5.1 – Mapeamento entre o dia da semana e o algarismo usado para preencher a planilha com horas de sobre aviso.....	33

LISTA DE ABREVIATURAS E SIGLAS

AJAX	Asynchronous Javascript and XML
API	Application Programming Interface
DE	Development Environment
IDE	Integrated Development Environment
J2EE	Java2 Platform Enterprise Edition
JDBC	Java Database Connectivity
JSF	JavaServer Faces
JSP	JavaServer Pages
KB	Knowledge Base
MVC	Model View Controller
Nexp	Número de plantonistas experientes
Nini	Número de plantonistas iniciantes
Nint	Número de plantonistas intermediários
Nsem	Número de semanas
PO	Product Owner
PROD	Production Environment
SMS	Short Message Service
SQL	Structured Query Language
STAGING	Test Environment
SVN	Subversion
UI	User Interface
URL	Uniform Resource Locator
XML	Extensible Markup Language

SUMÁRIO

1 INTRODUÇÃO.....	10
2 O SUPORTE A PRODUÇÃO.....	11
3 TECNOLOGIAS E TRABALHOS RELACIONADOS.....	13
3.1 JavaServer Faces.....	13
3.2 Primefaces.....	14
3.3 Hibernate.....	14
3.4 MySQL.....	15
3.5 Apache Tomcat Server.....	15
3.6 Tortoise SVN.....	16
3.7 JUnit.....	16
3.8 EclEmma.....	17
3.9 Trabalhos relacionados.....	17
3.9.1 Opsweekly.....	18
3.9.2 iPlantão.....	18
3.9.3 Pagerduty.....	18
4 ONCALL MANAGER, O DESENVOLVIMENTO DO SOFTWARE.....	20
4.1 O processo de desenvolvimento ágil.....	21
4.2 Scrum gerenciando o desenvolvimento do OnCall Manager.....	21
4.2.1 Gerencia do backlog do produto utilizando Pivotal Tracker.....	22
4.2.2 OnCall Manager e as sprints de desenvolvimento.....	23
5 ESPECIFICAÇÃO FUNCIONAL DA APLICAÇÃO.....	25
5.1 Controle de contas de usuários.....	25
5.2 Controle e geração da escala de plantão.....	26
5.3 Controle dos Acionamentos.....	29
5.4 Controle da Knowledge Base.....	30
5.5 Controle de Checklist.....	31
5.6 Controle de Agenda/Contatos.....	32
5.7 Módulo de Geração de planilha financeira.....	32
6 CONCLUSÃO.....	35
6.1 Trabalhos futuros.....	35
REFERÊNCIAS.....	36

1 INTRODUÇÃO

Uma equipe de suporte a produção de software é responsável por resolver incidentes que ocorrem em um ambiente do software onde as funcionalidades estão disponíveis a usuários clientes do produto suportado.

O sistema a ser desenvolvido, chamado OnCall Manager, tem por objetivo informatizar um conjunto de processos da equipe de suporte, ou seja, resolver o problema de gerar a escala de plantonistas da equipe de suporte, resolver o problema da geração de escala de pagamento de membro da equipe de suporte quando eles estiverem de plantão, servir de ferramenta de controle de acionamentos que ocorreram durante os chamados aos plantonistas e, também, servir como apoio para encontrar soluções incidentes e acionamentos no software suportado.

A estrutura do trabalho está dividida em duas grandes áreas principais, a primeira parte engloba os capítulos 2 e 3 e apresentam o problema a ser resolvido, ou seja, a informatização dos processos manuais da equipe de suporte, descrevem as tecnologias utilizadas para tornar as funcionalidades do sistema disponíveis para os usuários e algumas aplicações que resolvem problemas similares. A segunda parte engloba os capítulos 4 e 5 e apresenta a solução para o problema descrito na primeira parte, contém detalhes de como os requisitos levantados analisando o problema foram organizados em uma metodologia de desenvolvimento para virarem funcionalidades no sistema, a outra parte descreve o funcionamento dos módulos da aplicação desenvolvida e as regras que as funcionalidades devem respeitar. Por fim temos uma análise dos resultados obtidos e alguns trabalhos futuros que podem ser implementados na aplicação.

2 O SUPORTE A PRODUÇÃO

Gerenciar uma equipe de suporte a produção de um portal web de uma empresa não é uma tarefa simples, existem inúmeras atividades que tal equipe deve desempenhar para manter a aplicação acessível aos clientes durante o maior período possível.

Entre as responsabilidades da equipe de suporte estão prover análise e soluções a curto ou longo prazo para problemas reportados nas aplicações suportadas encontrando a causa raiz do erro e não somente prover uma solução paliativa, propor melhorias nas aplicações suportadas baseado nas expectativas dos clientes, prover suporte técnico todos os dias, incluindo feriados e finais de semana durante as 24 horas do dia, monitorar servidores onde a aplicação está instalada, monitorar alertas e escalar incidentes encontrados no sistema, desenvolver aplicações para automatizar processos do time, criar relatórios sobre os dados da aplicação, assegurar que a funcionalidade e usabilidade da aplicação suportada está alinhada com as regras de negócio da organização, ou seja, garantir que uma funcionalidade de fato obedeça as suas especificações.

Já que falhas em sistemas de computação são inevitáveis, a precisão e velocidade na detecção da causa de um erro e a correção desses erros é fundamental para a satisfação do cliente portanto podemos analisar o suporte ao ambiente de produção de um software com um pós venda do produto responsável por reforçar a confiabilidade no produto suportado.

Apesar de todo o controle da qualidade de software instalado em um ambiente de produção de um portal que possui centenas de milhares de clientes e milhões de usuários cadastrados capazes de acessar o site, é esperado uma grande quantidade de incidentes reportados por esses usuários. Na teoria, a quantidade de erros e problemas detectados pelos clientes é diretamente proporcional à soma do número de acessos de usuários dos clientes, ou seja, quanto mais acessos ao site maior a quantidade de incidentes que o suporte deve analisar e resolver.

Isso de fato é perceptível se imaginarmos que, por exemplo, um defeito no código da aplicação não bloqueia o duplo clique de um botão. Se esse botão ao ser clicado dispara um evento que incrementa o salário de um empregado, um duplo clique involuntário irá gerar um duplo aumento de salário, ou seja, um estado inconsistente do ponto de vista do usuário. Quanto mais usuários acessarem essa funcionalidade defeituosa de aumento de salário maior a chance de o defeito ocorrer e maior será a quantidade de incidentes abertos para a equipe de suporte requisitando a correção desse duplo aumento.

Uma empresa deve garantir que o esforço de trabalho dos funcionários membros da

equipe de suporte deve estar voltada para a correção dos erros do software. Isso justifica uma ferramenta que substitua os processos gerenciais do time que são manuais por processos informatizados. O sorteio da escala de plantão através de papéis com os nomes dos empregados não é uma maneira justa e eficaz de gerar esta escala da mesma forma, não é eficaz que um gerente da equipe de suporte tenha que manualmente gerar e verificar quantas horas cada plantonista esteve escalado em um determinado período.

O software a ser suportado pode ainda possuir várias versões diferentes de código, cada versão possui sua arquitetura, cada versão possui equipes de desenvolvimento independentes, cada versão pode possuir diferentes tecnologias e o efeito colateral disso é a necessidade de cada plantonista conhecer as particularidades das versões afim de garantir um suporte de qualidade. Uma base de consulta rápida e de fácil acesso sobre as particularidades de cada versão de código é outra funcionalidade necessária.

Quando a base de consulta não conseguir contribuir para resolução de um incidente é necessário ter uma agenda disponível com contatos das equipes responsáveis pelo desenvolvimento da versão defeituosa, essa agenda deve conter informações atualizadas e que possa ser modificada dinamicamente. O apoio de um gerenciador computacional que possui tais funcionalidades, não se limitando a essas, tem como objetivo aumentar a produtividade da equipe de suporte em termos da quantidade de incidentes resolvidos.

3 TECNOLOGIAS E TRABALHOS RELACIONADOS

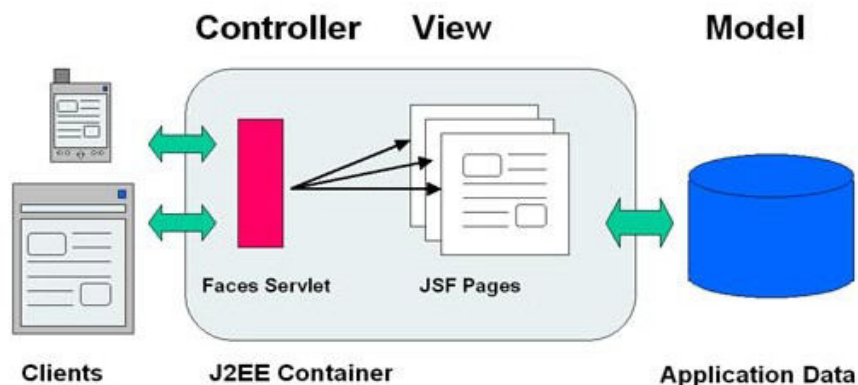
O software OnCall Manager estará disponível para os usuários através de acesso a uma URL, ou seja, é um programa acessível pelo *web browser*. Para que o software de fato seja instalado em um servidor e acessível aos usuários, foram utilizadas algumas ferramentas no desenvolvimento para aumentar a produtividade do desenvolvedor. As sessões a seguir descrevem quais foram essas ferramentas e apresentam as características que justificam a sua utilização no desenvolvimento do produto. Além disso são apresentados alguns trabalhos relacionados, ou seja, algumas ferramentas que porvêm soluções para problemas similares.

3.1 JavaServer Faces

JavaServer Faces mudou a forma que aplicações web, baseadas em Java são codificadas, desenvolvido para simplificar a criação de interfaces de usuário para aplicações de alta performance, que frequentemente é a parte mais difícil e intediante do desenvolvimento de aplicações web. O *framework* JSF provê componentes vizuais pré prontos para construir interfaces web melhorando a produtividade e a facilidade no desenvolvimento, Fornece um conjunto de tags JSP para acessar os componentes, facilita o acesso e a manipulação dos dados de aplicação da interface de usuário web. O JSF automaticamente manipula o estado da interface de usuário entre múltiplas requisições e multiplos clientes de uma forma simples e discreta, ele é um *framework* de desenvolvimento que é amigável para desenvolvedores com diferentes habilidade (BURNS & SCHALK, 2010).

O JSF segue o padrão arquitetural MVC (*Model-View-Controller*), a idéia do padrão MVC é dividir uma aplicação em três camadas: modelo, visualização e controle. O modelo é responsável por representar os objetos de negócio, manter o estado da aplicação e fornecer ao controlador o acesso aos dados. A visualização representa a interface com o usuário, sendo responsável por definir a forma como os dados serão apresentados e encaminhar as ações dos usuários para o controlador. Já a camada de controle é responsável por fazer a ligação entre o modelo e a visualização, além de interpretar as ações do usuário e as traduzir para uma operação sobre o modelo, onde são realizadas mudanças e, então, gerar uma visualização apropriada. Se mapearmos o MVC para o JSF temos as páginas com os componentes JSF como a view, o *FacesServlet* que é responsável por receber requisições da Web, redicioná-las para o modelo e então remeter uma resposta faz o papel do controlador e as entidades Java fazem o papel do modelo. A figura 3.1 mostra o diagrama da arquitetura JSF MVC.

Figura 3.1 – Arquitetura MVC do JavaServer Faces



Fonte: <http://www.javabeat.net/jsf-2/>

3.2 Primefaces

PrimeFaces é um componente JSF de código aberto com várias extensões e um grande conjunto de componentes tais quais HtmlEditor, Dialogs, Charts que contribuem para tornar a interface do usuário mais atrativa. É baseado no padrão de APIs Ajax JSF, ele é um *framework* leve que não necessita de configurações e não possui dependências, possui um kit UI para criação de aplicações mobile, é extremamente fácil utilizar os componentes do Primefaces no produto desenvolvido. Além disso o *framework* possui uma vasta documentação, uma larga e ativa comunidade e isso é importante no caso de encontrar solução para bloqueios e erros relacionados a não funcionamento de componentes Primefaces.

3.3 Hibernate

Trabalhar com software orientado a objetos e base de dados relacionais pode ser pesado e pode consumir muito tempo de desenvolvimento. O custo de desenvolvimento é significativamente alto tendo em vista a diferença entre como os dados são representados nos objetos versus as bases de dados relacionais. As bibliotecas do Hibernate são uma solução de mapeamento Objeto/Relacional para ambientes Java, o termo Objeto/Relacional refere se a técnica que mapeia dados de uma representação de um modelo de objeto Java com uma representação de modelo de dados relacional (e vice-versa).

Além de controlar o mapeamento de classes Java para tabelas de banco de dados (e de tipos de dados Java para tipos de dados SQL) o hibernate também provê facilidades para recuperação dos dados. Hibernate pode significativamente reduzir o tempo de

desenvolvimento que seria gasto com a manipulação manual dos dados com SQL e JDBC. A figura 3.2 mostra um trecho do arquivo persistence.xml, esse arquivo possui a configuração e o mapeamento entre a classe Java e uma entidade do banco de dados.

Figura 3.2 – Parte do código do persistence.xml

```
<persistence-unit name="user">
    <provider>org.hibernate.ejb.HibernatePersistence</provider>
    <class>com.hermanyg.modelo.Usuario</class>
    <properties>
        <property name="javax.persistence.jdbc.driver" value="com.mysql.jdbc.Driver" />
        <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/oncall_staging" />
        <property name="javax.persistence.jdbc.user" value="root" />
        <property name="javax.persistence.jdbc.password" value="root" />
        <property name="hibernate.dialect" value="org.hibernate.dialect.MySQLDialect" />
        <property name="hibernate.show_sql" value="true" />
        <property name="hibernate.format_sql" value="true" />
        <property name="hibernate.hbm2ddl.auto" value="update" />
    </properties>
</persistence-unit>
```

Fonte: Imagem criada pelo autor

3.4 MySQL

Os dados armazenados pela aplicação OnCall Manager são modelados utilizando o modelo relacional, ou seja, esses dados são percebidos como tabelas e relações entre tabelas. A gerência dos dados manipulados pela aplicação do projeto é feita utilizando o MySQL, um servidor gratuito, com código aberto, confiável, de alta performance que é fácil de usar e de configurar.

Existe ainda a ferramenta MySQL Workbench que é uma ferramenta gráfica que disponibiliza um abrangente conjunto de ferramentas para configurar o servidor de banco de dados, possui ainda ferramentas para aumentar a produtividade no desenvolvimento de *scripts* SQL e modelagem dos dados, essa ferramenta foi utilizada no desenvolvimento do produto.

3.5 Apache Tomcat Server

O Tomcat é um servidor para aplicações Java web e é um software de uso livre tanto para aplicações comerciais quanto para aplicações não comerciais. Uma importante funcionalidade do servidor é o Tomcat Manager Web Application que provê uma funcionalidade básica para gerenciar aplicações web rodando em um servidor Tomcat de qualquer browser, algumas dessas funcionalidades incluem a habilidade de instalar, iniciar, parar e remover aplicações web (VUKOTIC & GOODWILL, 2011). O Tomcat permite ainda

a configuração da aplicação web, ou seja, através do Tomcat podemos configurar o tempo de duração de cada sessão criada na aplicação. O OnCall Manager está instalado em um servidor Tomcat versão 7.

3.6 Tortoise SVN

SVN, ou Subversion, é um sistema de controle de versão de código e é útil em projetos ágeis onde mudanças no código fonte da aplicação são comuns. O OnCall Manager tem 3 versões de código no SVN, a versão DE que é o ambiente de desenvolvimento onde as mudanças são adicionadas e testadas pelo desenvolvedor do projeto e é a mais volátil, a versão STAGING que é o ambiente que recebe o código modificado e aceito no ambiente de desenvolvimento, o ambiente STAGING é usado para demonstrar as funcionalidades para o PO do projeto e o projeto tem ainda a versão PROD, o código aceito e desenvolvido no ambiente de desenvolvimento e aceito no ambiente STAGING é adicionado na versão PROD, essa versão possui o código que será instalado no ambiente de produção do produto, é o ambiente que os usuários acessam.

O Tortoise é uma ferramenta gráfica que é responsável por facilitar a configuração e manipulação desses múltiplos ambientes e versões do produto. É integrado com o sistema operacional Windows e os padrões de exploração dos arquivos no Tortoise seguem o padrão Windows.

3.7 JUnit

Uma parte importante do desenvolvimento de um projeto é garantir a qualidade do software a ser desenvolvido. Do ponto de vista de um usuário a qualidade do software é medida através da facilidade do uso, do desempenho e da confiabilidade dos resultados, o teste do software ajuda a garantir a confiabilidade desses resultados, ou seja, testando as funcionalidades implementadas e garantindo que de fato elas satisfazem as especificações do cliente e verificando se o código não apresenta erros que podem gerar falhas na hora do uso do software aumentam a qualidade do software entregue para os usuários.

Testar o código fonte é um passo do teste de um software e como a lógica de negócio do OnCall Manager é implementada em Java foi utilizado o JUnit para realização de testes unitários, testes unitários são executados sobre os métodos implementados nas classes de um projeto.

O *framework* JUnit possui uma vasta documentação e uma comunidade de desenvolvedores ativa, os testes executam e os resultados dos testes são fornecidos rapidamente, é possível testar uma parte específica da aplicação e o JUnit é livre. Após executar testes no OnCall Manager foram detectados alguns erros de codificação, por exemplo, os métodos `getAlterarSecundarioTest` e `onCellEdit` dentro da classe `ConsultaEscalaBean` disparavam a exceção `java.lang.ArrayIndexOutOfBoundsException` quando nenhuma escala de plantão existe no banco de dados, essa exceção é disparada quando tentamos acessar uma variável vetor em um índice ilegal.

3.8 EclEmma

EclEmma é uma ferramenta gratuita para a IDE Eclipse as suas principais características são a capacidade de analisar o critério de cobertura de código para os testes criados com JUnit e a ferramenta não exige modificação do projeto desenvolvido. O relatório gerado a partir do EclEmma mostra que os testes unitários desenvolvidos usando o JUnit cobriram 70% do código desenvolvido no OnCall Manager, a figura 3.3 mostra quais as classes com métodos mais testados no projeto.

Figura 3.3 – Relatório de cobertura de testes gerado pela ferramenta EclEmma

Element	Coverage	Covered Instructio...	Missed Instructions	Total Instructions
PIDSonCall	70,0 %	4.449	1.904	6.353
src	70,0 %	4.449	1.904	6.353
com.hermanyg.Bean	70,0 %	3.262	1.399	4.661
ConsultaEscalaBean.java	9,8 %	43	394	437
GeraEscalaBean.java	85,5 %	1.170	199	1.369
EditaUsuarioBean.java	3,6 %	6	160	166
UsuarioBean.java	0,0 %	0	154	154
NavegacaoBean.java	0,0 %	0	86	86
LoginControlBean.java	17,3 %	17	81	98
GeraPlanilhaBean.java	95,3 %	1.390	68	1.458
TesteListaUsuariosBean.java	0,0 %	0	52	52
RegistraAcionamentoBean.java	53,8 %	49	42	91
ListaUsuariosBean.java	75,8 %	122	39	161
ListaAcionamentosBean.java	86,7 %	229	35	264
LoggedUserBean.java	0,0 %	0	31	31
ListaKbBean.java	52,5 %	31	28	59
RequisitoBean.java	86,6 %	116	18	134
CriaUsuarioBean.java	91,8 %	67	6	73
RegistraKbBean.java	78,6 %	22	6	28
com.hermanyg.modelo	43,5 %	137	178	315
Pacotedeteste	80,7 %	562	134	696
com.hermanyg.Filter	0,0 %	0	115	115
com.hermanyg.Dao	86,2 %	488	78	566
UsuarioDao.java	76,4 %	252	78	330
DocumentoDAO.java	100,0 %	99	0	99
RequisitoDAO.java	100,0 %	137	0	137

Fonte: Imagem criada pelo autor

3.9 Trabalhos relacionados

As escalas de plantão são uma prática comum em vários setores no mercado profissional, a forma como esse problema é resolvido varia de acordo com as necessidades e

complexidades de cada escala. Atualmente os exemplos de software gerenciadores de escalas são inúmeros e algumas se aproximam das necessidades de uma equipe de suporte de software, porém podem não atender as particularidades de cada equipe.

3.9.1 Opsweekly

O projeto desenvolvido pela empresa norte americana Etsy, Inc denominado Opsweekly, cujo código fonte encontra-se em <https://github.com/etsy/opsweekly>, é uma ferramenta que tem por objetivo centralizar e organizar o trabalho de uma equipe de suporte bem como ajudar a entender e incrementar a rotação de plantão através de uma pesquisa de plantão.

Cada membro da equipe pode escrever um “status” semanal para informar o time sobre o que ele esteve trabalhando, oferece ainda um sistema de classificação de alertas de acordo com a gravidade do problema encontrado no software suportado e gera reportes sobre estes alertas.

3.9.2 iPlantão

Essa ferramenta brasileira disponibiliza serviços na área da medicina e não na área da informática, apesar disso, alguns módulos podem ser utilizados para resolver a gerencia de uma equipe de suporte a software, por exemplo, a ferramenta gerencia a escala de plantão, provê a comunicação entre serviços, setores e equipes que interagem entre si e também organiza os dados para os pagamentos dos profissionais plantonistas.

3.9.3 Pagerduty

Pagerduty (<https://www.pagerduty.com/tour/on-call-schedule/>) é um software que possui um módulo que permite criar multiplas escalas de plantão, oferece uma visibilidade instantanea de quem está de plantão, o software se encarrega de notificar por e-mail ou SMS quando o usuário estará de plantão. Também oferece o emprego de uma política de escalagem quando um incidente é crítico, por exemplo, quando um alerta crítico é lançado no sistema o alerta além de notificar o plantonista também notifica o gerente e o diretor do projeto.

Apesar de existirem tais aplicações que resolvem alguns problemas elas não suficientes para um time de supore. O comparativo da tabela 3.1 justifica o desenvolvimento

do OnCall Manager baseado nos requisitos do software.

Tabela 3.1 – Tabela comparativa entre funcionalidades das aplicações

	OnCall Manager	Opsweekly	iPlantão	Pagerduty
Gerar Escala de plantão	Possui	Possui	Possui	Possui
Gerar Escala de plantão baseada em categoria de usuários	Possui	Não possui	Possui	Não possui
Gerar Planilha para setor financeiro	Possui	Não possui	Não possui	Não possui
Criação de KBs	Possui	Não possui	Não possui	Não possui
Controle de conta de usuários	Possui	Possui	Possui	Possui
Controle de acionamentos	Possui	Possui	Não possui	Possui
Agenda de contatos	Possui	Não possui	Possui	Possui

Fonte: Tabela criada pelo autor

4 ONCALL MANAGER, O DESENVOLVIMENTO DO SOFTWARE

O começo do desenvolvimento de um software é o surgimento da ideia ou necessidade de informatizar um problema enfrentado pelas pessoas, transformar um processo manual que frequentemente apresenta erros de execução em um processo automatizado e controlado. Após ser bem testado o processo informatizado além de ser mais confiável também apresenta resultados com maior velocidade em relação ao processo manual e, visto a grande quantidade de hardwares móveis que existem nos dias atuais e que conseguem computar problemas de alta complexidade, a informatização aumenta a acessibilidade do processo responsável pela resolução do problema.

Se considerarmos como problema, por exemplo, a necessidade de um diretor de uma empresa aprovar o aumento de salário de alguns funcionários, o processo manual de assinar papéis aprovando essas requisições é lento comparado ao processo informatizado onde o gerente somente precisaria clicar em um botão de aprovação. Imagine ainda que o software possa enviar mensagens para o *smartphone* do gerente e que esse gerente possa fazer a aprovação dos aumentos salariais pelo próprio *smartphone*, a acessibilidade da ferramenta no dispositivo móvel contribui para o aumento da velocidade com que o problema é resolvido.

Para fazer com que a ideia inicial seja de fato implementada e disponibilizada para o usuário em forma de software são levantados os requisitos necessários do programa, esses requisitos são alavancados através de reuniões entre a empresa responsável pelo desenvolvimento do software e a empresa cliente. No caso do OnCall Manager os requisitos do sistema de gerenciamento de equipes de suporte a produção de um portal web foram definidos por um gerente de uma equipe de suporte em reuniões de análise de requisitos.

Com base nos requisitos do sistema é definido um processo de desenvolvimento do software capaz de informatizar estes requisitos. Um processo é definido como um coleção de atividades, ações e tarefas que são performadas quando um produto será criado. Cada uma dessas atividades, ações e tarefas encontra-se dentro de um *framework* ou modelo que define as suas relações com o processo e entre si. Um importante aspecto do processo de software é o fluxo do processo que descreve como as atividades, ações e tarefas que acontecem dentro de cada atividade do *framework* são organizadas em relação a sequência e tempo (PRESSMAN, 2000).

4.1 O processo de desenvolvimento ágil

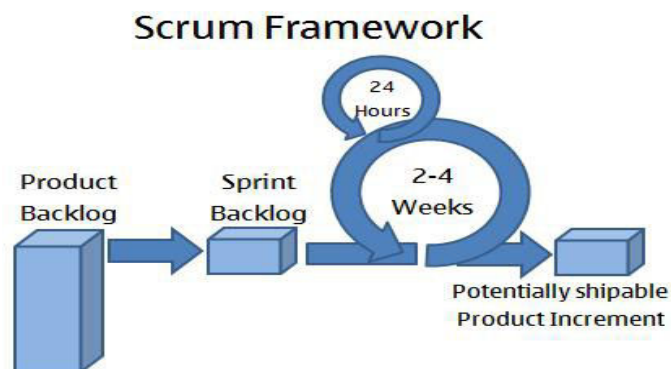
Agilidade tornou-se um chavão quando descrevemos um processo de desenvolvimento de software moderno. Todo mundo é ágil. Um time ágil é aquele capaz de responder apropriadamente a mudanças. Mudança é o que o desenvolvimento de software é. Mudanças no software a ser construído, mudanças nos membros do time, mudanças por causa de novas tecnologias, mudanças de todos os tipos que possam ter um impacto no produto desenvolvido. Suporte para mudanças deve estar embutido em tudo que é feito no software, algo que abraçamos porque ela é o coração e a alma do software. Um time ágil reconhece que um software é desenvolvido por individualidades trabalhando em times e onde as habilidades dessas pessoas estão no núcleo do sucesso do projeto (JACOBSON, 2002).

Existem alguns *frameworks* que gerenciam o processo de desenvolvimento de software que aceitam essas mudanças que ocorrem durante o processo e que tem por objetivo aumentar a produtividade do projeto, o Scrum é um desses frameworks.

4.2 Scrum gerenciando o desenvolvimento do OnCall Manager

Segundo o guia do scrum, ele é um *framework* dentro do qual pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível. O Scrum é leve, simples de entender e difícil de dominar (SCHWABER & SUTHERLAND, 2013). As funcionalidades levantadas a partir dos requisitos do software a ser desenvolvido são mantidas em uma lista chamada *product backlog*, elas então são priorizadas e adicionadas a outra lista, o *sprint backlog*, as histórias que fazem parte desta nova lista serão selecionadas pela equipe de desenvolvimento e implementadas dentro da *sprint*. Uma *sprint* normalmente tem duração de 2 a 4 semanas, variando de acordo com o projeto, lembre-se porém que uma vez definida a duração de cada *sprint* no início do projeto ela não é alterada. As histórias implementadas durante a *sprint*, após serem aceitas pelo *product owner*, incrementam o produto desenvolvido. Pressman (2010) definiu um fluxo de processo evolutivo como um processo que executa as atividades em modo circular e o framework scrum se encaixa nessa categoria já que o processo descrito acima se repete ciclicamente. A figura 4.1 ilustra o fluxo de execução do framework scrum.

Figura 4.1 – Fluxo de execução do framework scrum



Fonte: <http://www.expertprogrammanagement.com>

O time scrum é dividido em três papéis, o *product owner*, o time de desenvolvimento e o *scrum master*. O PO é responsável por maximizar o valor do produto e do trabalho do time de desenvolvimento e é a única pessoa responsável por gerenciar o *Backlog* do produto. No caso do OnCall Manager o PO era o gerente da equipe de suporte de uma empresa e de fato ele é o responsável pelo controle do backlog do produto. As atividades do PO consistiam em priorizar e organizar as estórias que seriam desenvolvidas na próxima *sprint*, garantir que o desenvolvedor entendesse os itens do *backlog* e aceitar as estórias entregues ao final de cada ciclo de desenvolvimento.

O Scrum foi adaptado para o desenvolvimento do OnCall Manager tendo em vista que as funções dos papéis do *scrum master* e do time de desenvolvimento foram somados e são de responsabilidade do desenvolvedor do software. O desenvolvedor, como time de desenvolvimento de apenas um integrante, possui as habilidades necessárias para transformar as estórias em funcionalidades do sistema, incrementando o produto.

Como *scrum master* o desenvolvedor teve que encontrar formas de resolver impedimentos no desenvolvimento do software, ajudar as partes interessadas a compreender e tornar aplicável o scrum e encontrar técnicas para o gerenciamento do *backlog*, a saída utilizada para o gerenciamento do *backlog* do OnCall Manager foi criar uma conta de usuário conjunta com o PO na ferramenta Pivotal Tracker.

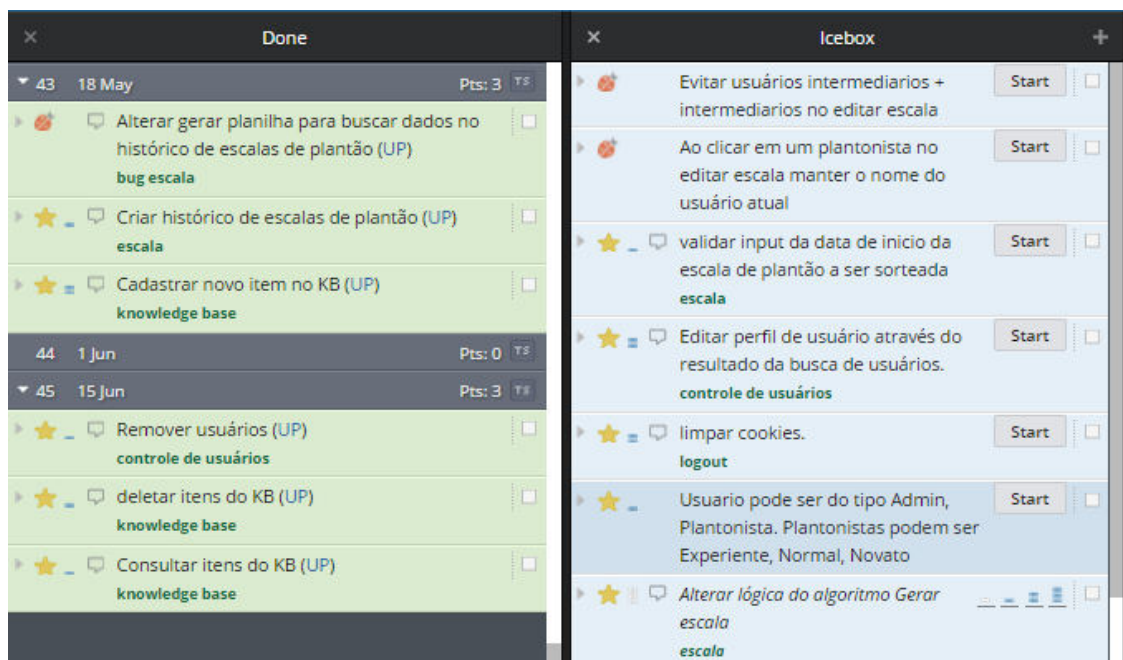
4.2.1 Gerencia do backlog do produto utilizando Pivotal Tracker

Para gerenciar o *backlog* do OnCall Manager e o *backlog* de cada *sprint* foi utilizada o Pivotal Tracker que é uma ferramenta que simplifica a priorização das estórias de usuários, a sua interface limpa dá uma boa visão sobre o status global do projeto, ela facilitou a adoção

de processo ágil no projeto. É fácil estimar as estórias na ferramenta e com essa estimativa é possível estabelecer expectativas e velocidade de desenvolvimento do time.

Após criar uma conta e compartilhar o acesso com todos os interessados no projeto as estórias foram criadas no *backlog* do produto e a partir daí elas eram priorizadas e incluídas em *sprints* de desenvolvimento. A necessidade de compartilhar as credenciais de acesso está ligada ao fato de que a criação de projetos gratuitamente na ferramenta está condicionada a esse projeto possuir apenas um membro. A medida que a implementação de uma estória é iniciada e seu status é alterado essa mudança é refletida no *pivotal tracker*, por exemplo, quando a estória é aceita após a demonstração da funcionalidade para o PO ela muda de *Delivered* para *Accepted* e vai para a lista *Done*, a lista de estórias finalizadas e aceitas.

Figura 4.2 – Estórias no backlog do produto e lista de estórias aceitas



Fonte: Imagem criada pelo autor

4.2.2 OnCall Manager e as sprints de desenvolvimento

A duração de cada sprint no desenvolvimento do OnCall Manager eram de 2 semanas, ao final de cada *sprint* era realizada um reunião retrospectiva com o PO onde eram apresentadas as funcionalidades implementadas discutindo se elas realmente estavam de acordo com as expectativas do cliente. Essa mesma reunião retrospectiva era também usada como um reunião de planejamento da próxima *sprint*, dúvidas do desenvolvedor sobre as estórias da próxima *sprint* eram sanadas pelo PO do software. A tabela 4.1 mostra as

funcionalidades implementadas em cada *sprint* de desenvolvimento.

Tabela 4.1 – Estórias implementadas em cada sprint

Sprint	Estórias Implementadas
0	ID 57313674 - Criar novos usuários ID 57314156 - Editar perfil de usuário ID 57320198 - Gerar a escala de plantão ID 58260604 - Garantir um usuário experiente em cada plantão ID 58260700 - Garantir que usuários novatos só podem ser plantonistas secundários
1	ID 58260128 - Login autenticado contra o cadastro de usuarios ID 60102800 - Gravar a escala sorteada no banco de dados ID 57320124 - Consultar a escala de plantão atual ID 60242212 - Adicionar opção de período de férias para plantões a serem escalados ID 61616572 - Criar mega menu ID 61626120 - Editar escala de plantão oficial
2	ID 57318316 - Criar requisito para checklist do plantonista ID 57317936 - Consultar o checklist de requisitos ID 57318214 - Deletar um requisito do checklist ID 57320062 - Registrar novo acionamento ID 57319962 - Listar todos os acionamentos ID 57327848 - Gerar planilha mensal com os plantonistas para o financeiro
3	ID 75691742 - Criar histórico de escalas de plantão ID 75691818 - Alterar gerar planilha para buscar dados no histórico de escalas de plantão ID 57315494 - Cadastrar novo item no KB
4	ID 57313780 - Remover usuários ID 57315110 - Consultar itens do KB ID 57316470 - Deletar itens do KB
5	ID 58260392 - Editar perfil de usuários como usuario admin ID 58260168 - Funcionalidades devem verificar a autenticação do usuário

Fonte: Tabela criada pelo autor

5 ESPECIFICAÇÃO FUNCIONAL DA APLICAÇÃO

O OnCall Manager está dividido em 7 módulos, o controle de contas de usuário, o controle e geração de escala de plantão, o controle de acionamentos, o controle do KB, o controle de checklist de requisitos, o controle da agenda de contatos e o módulo de geração da planilha para o departamento financeiro. As características de cada módulo estão nas seções a seguir e vão dar uma visão geral do funcionamento do software.

5.1 Controle de contas de usuários

Como o OnCall Manager possui várias funcionalidades e nem todos os usuários devem ter acesso a todas as funcionalidades disponíveis a ferramenta deve reconhecer tipos distintos de usuários e identificar quais funcionalidades devem estar disponíveis para o tipo de usuário logado. O controle de acesso é baseado em usuários do tipo administrador e usuários do tipo padrão. Os usuários padrão possuem um conjunto de funcionalidades menor em relação aos usuários do tipo administrador.

Um usuário administrador poderá criar outras contas de usuários, ele poderá criar tanto novos usuários administradores, marcando essa opção na tela de criação de usuários, quanto usuários padrão, além disso um usuário administrador poderá remover um usuário de qualquer tipo e poderá editar o perfil de um outro usuário incluindo a categoria (Experiente, Iniciante, Novato, sem categoria) do usuário e o tipo (administrador ou padrão) do usuário. Qualquer usuário poderá editar seu próprio perfil excluindo-se poder editar o tipo de usuário e a categoria.

Os usuários administradores devem ter acesso as páginas no diretório **/admin** e os usuários padrão devem ter acesso as páginas do diretório **/user**. A aplicação controla esses acessos usando uma configuração no web.xml do projeto e classes Java (*AdminPagesFilter* e *UserPagesFilter*) que implementam os filtros de acesso baseados no papel de cada usuário cadastrado.

Figura 5.1 – web.xml do projeto com a configuração de filtros para autenticação de acesso.

```

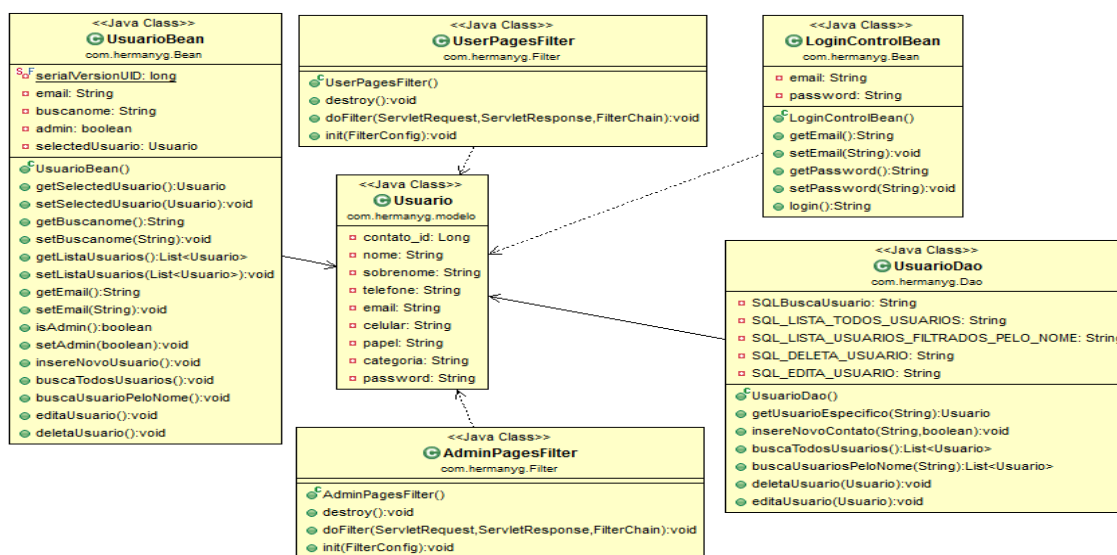
<filter>
  <filter-name>AdminFilter</filter-name>
  <filter-class>com.hermanyg.Filter.AdminPagesFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>AdminFilter</filter-name>
  <url-pattern>/admin/*</url-pattern>
</filter-mapping>
<filter>
  <filter-name>UserFilter</filter-name>
  <filter-class>com.hermanyg.Filter.UserPagesFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>UserFilter</filter-name>
  <url-pattern>/user/*</url-pattern>
</filter-mapping>

```

Fonte: imagem criada pelo autor

Do ponto de vista das estruturas de dados envolvidos nessa funcionalidade foram criadas as classes apresentadas no diagrama de classes da figura 5.2. Essas classes possuem todos os atributos e operações necessárias para a funcionalidade de controle de acessos.

Figura 5.2 – Diagrama de classes para o controle de acesso



Fonte: Imagem criada pelo autor

5.2 Controle e geração da escala de plantão

Além da carga de trabalho normal de um funcionário da equipe de suporte este trabalhador esta condicionado a fazer plantões semanais. Ao estar escalado como plantão o funcionário atende chamadas fora do horário comercial de trabalho, tais chamadas são comuns e incluem requisições de correções de erros de instalação de software, erros de

migração de clientes, erros de configuração da arquitetura do software suportado e outros incidentes.

Para estar apto a entrar na escala de plantão um usuário deverá estar em uma das seguinte categorias:

- Iniciante: plantonista com pouca experiência de suporte, com conhecimento básico na aplicação suportada.
- Intermediário: plantonista que possui alguma experiência em suporte com um bom conhecimento na aplicação suportada, capaz de agir com velocidade e corretude caso acionado.
- Experiente: plantonista que possui conhecimento pleno na aplicação suportada capaz de agir com velocidade e corretude caso acionado. Apto a identificar a causa de incidentes na aplicação suportada e corrigi-los bem como ter uma excelente compreensão das aplicações que interagem com o software suportado. Profissional com nível técnico elevado.

A cada semana dois plantonistas ficam de sobre aviso e responsáveis por atender eventuais chamados ao suporte, um deles é o plantão primário, responsável por atender as chamadas e um secundário, responsável por atender chamadas caso o primário esteja indisponível, por exemplo, atendendo outro chamado.

Quando uma nova escala irá ser gerada um usuário administrador selecionará uma data de início de sorteio e cada usuário candidato a plantonista poderá selecionar múltiplos períodos de ausência e que não poderá estar de plantão, por exemplo, durante as suas férias.

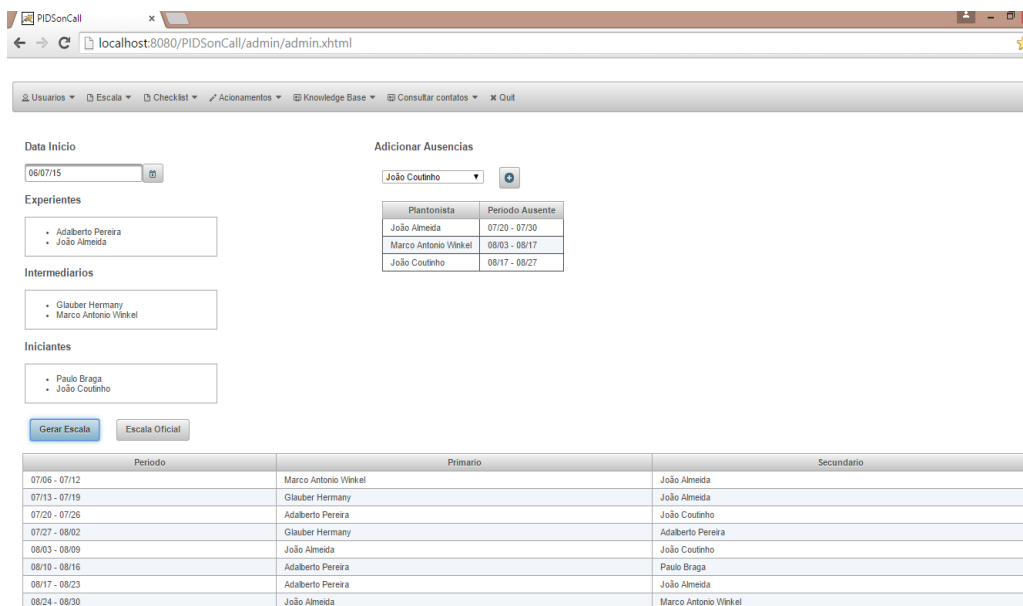
O software então deve considerar os períodos de ausência de cada usuário para gerar a próxima escala bem como não poderá sortear usuários iniciantes como plantonistas primários, deverá obedecer a regra de sempre sortear pelo menos um usuário experiente em cada semana de plantão, ou seja, as duplas de plantão possíveis são: primário experiente e secundário experiente, primário experiente e secundário intermediário, primário intermediário e secundário experiente e primário experiente e secundário iniciante.

Tendo em vista as regras acima a quantidade de semanas sorteadas obedecerá a equação,

$$N_{sem} = \frac{N_{exp}!}{2 \cdot (N_{exp} - 2)!} \cdot 2 + N_{int} \cdot N_{exp} \cdot 2 + N_{exp} \cdot N_{ini}$$

onde N_{sem} é o número de semanas totais a serem sorteadas, N_{exp} é o número de plantonistas experientes, N_{int} é o número de plantonistas intermediários e N_{ini} é o número de plantonistas iniciantes. A escala pode ser gerada inúmeras vezes até que uma seja escolhida e salva de fato como a próxima escala de plantão na base de dados fazendo com que a escala anterior seja movida para o histórico de escalas de plantão.

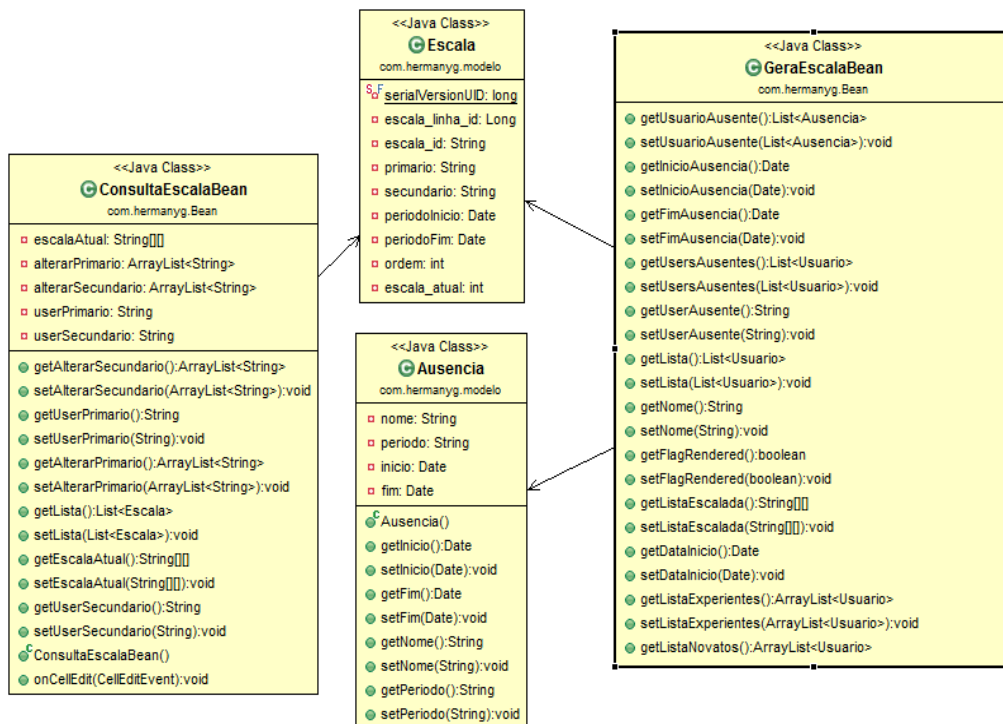
Figura 5.3 – Tela de geração da próxima escala de plantão



Fonte: Imagem criada pelo autor

Do ponto de vista das estruturas de dados envolvidos nessa funcionalidade foram criadas as classes apresentadas no diagrama de classes da figura 5.4. Essas classes possuem todos os atributos e operações necessárias para a funcionalidade de controle e geração de escalas de plantão.

Figura 5.4 – Diagrama de classes para o controle e geração de escalas de plantão



Fonte: Imagem criada pelo autor

5.3 Controle dos Acionamentos

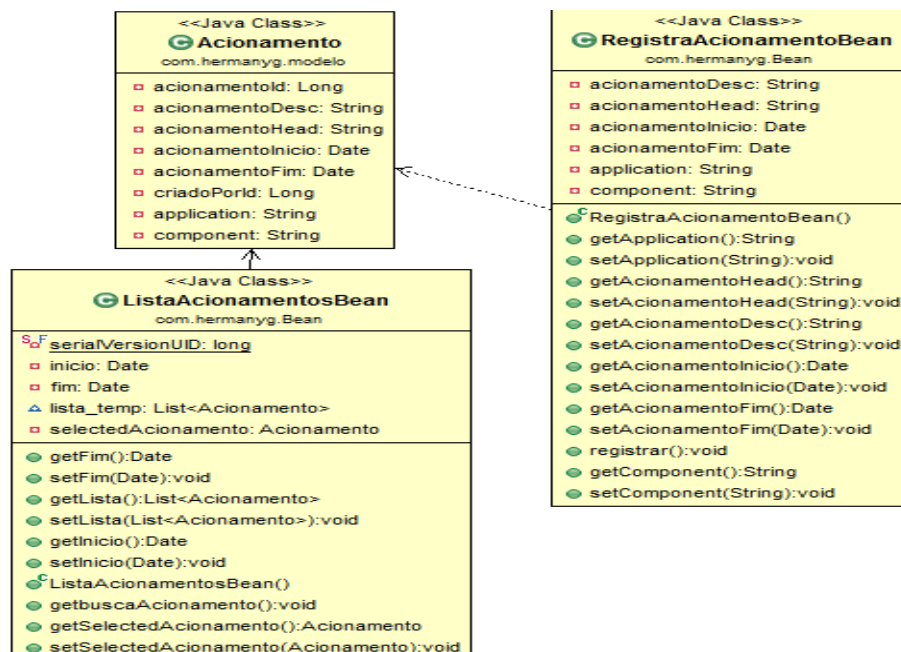
Um plantonista pode ser acionado a qualquer momento fora do horário comercial. Esses chamados são requisições de suporte para a aplicação suportada pelo time, são requisições para correção de erros de quaisquer natureza tais quais erros específicos em uma determinada página ou erros específicos para um cliente.

Quando um usuário é acionado ele deve poder cadastrar o acionamento no OnCall Manager inserindo título, descrição, hora de início e hora do fim do acionamento, resolução do acionamento, a aplicação afetada e o componente da aplicação impactada pelo erro se o plantonista achar essa informação relevante para a equipe de suporte.

A aplicação deverá oferecer um página com a lista de acionamentos registrados no banco de dados com a opção de filtrar acionamentos pela data de início ou data final dos acionamentos ou buscar acionamentos entre um data inicial e uma data final.

Do ponto de vista das estruturas de dados envolvidos nessa funcionalidade foram criadas as classes apresentadas no diagrama de classes da figura 5.5. Essas classes possuem todos os atributos e operações necessárias para a funcionalidade de controle de acionamentos.

Figura 5.5 – Diagrama de classes para o controle de acionamentos



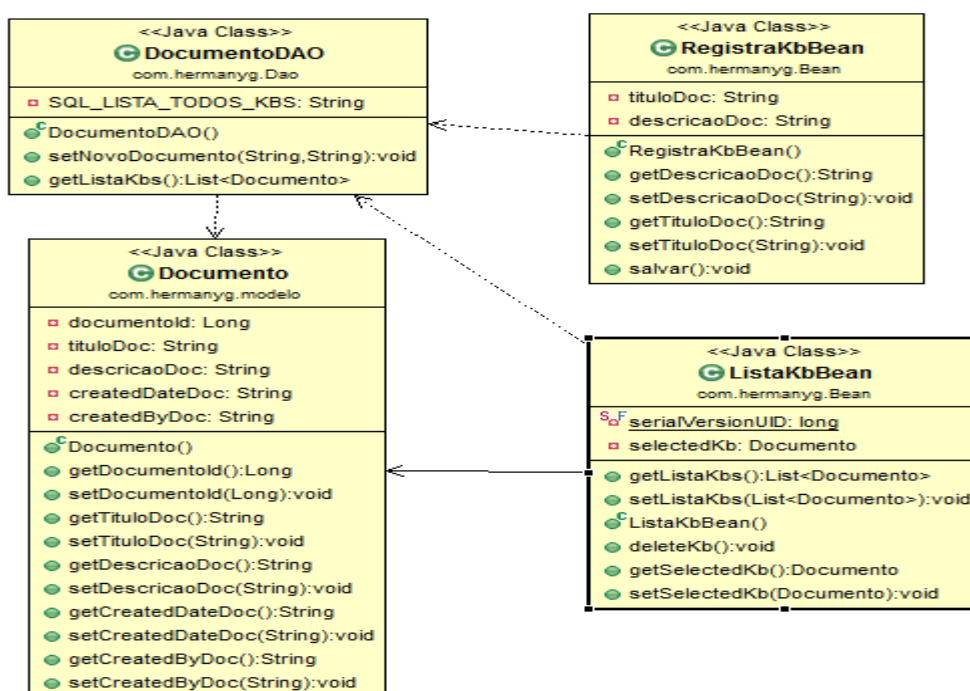
Fonte: Imagem criada pelo autor

5.4 Controle da Knowledge Base

Knowledge Base é um termo inglês e no escopo do OnCall Manager significa uma funcionalidade responsável por oferecer uma tela de cadastro de itens que possam servir como consulta para a resolução de eventuais acionamentos futuros. O cadastro de cada item é simples, o usuário seja ele administrador ou usuário padrão deve incluir um título para o item e uma descrição, essa descrição é um texto sem formatação. A aplicação deve oferecer também uma tela para consulta desses itens cadastrados na *Knowledge Base*. Exemplos de itens que a equipe considera úteis cadastrar são consultas *SQL*, comandos linux que fazem buscas em logs da aplicação suportada, comandos para chamadas de *webservices* entre outros.

Do ponto de vista das estruturas de dados envolvidos nessa funcionalidade foram criadas as classes apresentadas no diagrama de classes da figura 5.6. Essas classes possuem todos os atributos e operações necessárias para a funcionalidade de controle da knowledge base.

Figura 5.6 – Diagrama de classes para o controle da knowledge base



Fonte: Imagem criada pelo autor

5.5 Controle de Checklist

Cada usuário a plantonista deve satisfazer uma lista de pré requisitos chamado *checklist*, se um usuário não possuir conhecimentos nos requisitos do *checklist* ele não poderá ser escalado como plantonista. Essa lista de requisitos é dinâmica e um usuário administrador poderá remover ou criar um novo requisito e um usuário padrão poderá consultar a lista de requisitos. A figura 5.7 mostra a tela de consulta do *checklist* com a opção de deleção quando um usuário administrador está acessando o OnCall Manager.

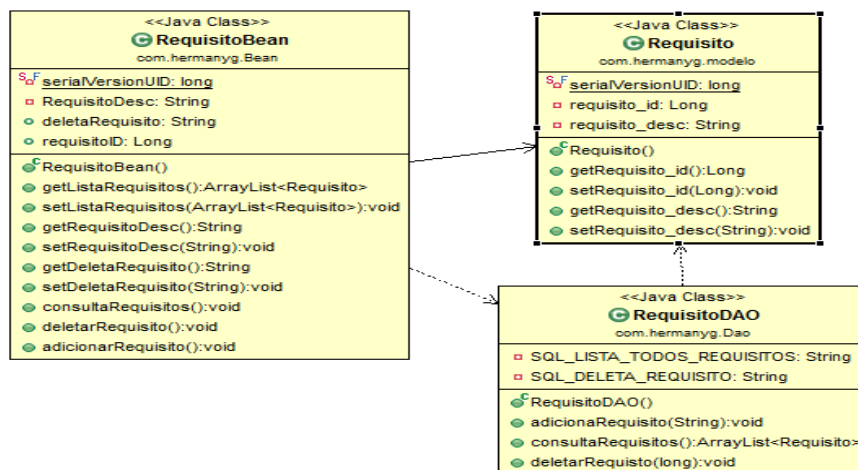
Figura 5.7 – Tela com requisitos necessários para um usuário estar apto a ser plantonista



Fonte: Imagem criada pelo autor

Do ponto de vista das estruturas de dados envolvidos nessa funcionalidade foram criadas as classes apresentadas no diagrama de classes da figura 5.8. Essas classes possuem todos os atributos e operações necessárias para a funcionalidade de controle do checklist.

Figura 5.8 – Diagrama de classes para Controle do checklist



Fonte: Imagem criada pelo autor

5.6 Controle de Agenda/Contatos

A equipe de suporte interage com diversas equipes dentro da empresa, equipes de desenvolvimento, equipes de arquitetura de software, equipes de gerência e direção, equipes de outras aplicações da empresa. O OnCall Manager deve oferecer telas para consulta de endereço de *e-mails*, telefones de funcionários da empresa responsáveis por determinado componente da aplicação suportada, endereço de *e-mails* de times que interagem com a equipe de suporte. Essa informação é adicionada pelo desenvolvedor do OnCall Manager de acordo com as necessidades da equipe de suporte e requisições do gerente da equipe de suporte.

5.7 Módulo de Geração de planilha financeira

Como envolve dados financeiros esse módulo do OnCall Manager é sensível e erros são desagradáveis. A planilha deve ser enviada para o departamento financeiro através de carta e por isso a planilha deve ser impressa em papel. O papel do software gerenciador OnCall Manager é disponibilizar um botão de download da planilha, a planilha deve incluir as horas de sobre aviso dos plantonistas primários e secundários durante um período de 1 mês a contar do dia 20 de um mês até o dia 19 do mês seguinte, por exemplo, se o gerente da equipe quiser gerar a planilha com as horas de sobre aviso do período Março a Abril de 2015 então esse gerente poderá selecionar esse período na tela e o algoritmo que vai gerar a planilha deve buscar os plantonistas do período de 20 de Março de 2015 a 19 de Abril de 2015.

O algoritmo que irá gerar a planilha deve preencher 6 campos no arquivo a ser gerado, esses campos são requisitos do departamento financeiro da empresa e são a data do plantão no formato DD/MM/AAAA onde cada D é um algarismo e formam o dia do mês, cada M é um algarismo e formam o mês do ano e cada A é um algarismo e formam o ano do plantão, a planilha deve conter ainda o campo competência que é preenchido com o mês do plantão escrito por extenso. O próximo campo da planilha é um algarismo que representa o dia da semana e ele é preenchido obedecendo o mapeamento da tabela 5.1.

Tabela 5.1 – Mapeamento entre o dia da semana e o algarismo usado para preencher a planilha com horas de sobre aviso

Dia da Semana	Algarismo correspondente
Domingo	1
Segunda-feira	2
Terça-feira	3
Quarta-feira	4
Quinta-feira	5
Sexta-feira	6
Sábado	7

A planilha ainda deverá conter o campo nome do plantonista do dia, o campo motivo que é sempre preenchido com “Plantão Suporte a Produção” e a quantidade de horas de sobre aviso preenchido com o valor 24 caso o dia da semana do plantão é sábado ou domingo e com o valor 16 caso contrário. Por fim a planilha deve mostrar a soma de horas de sobre aviso para cada plantonista no período. Essa funcionalidade está disponível somente para usuários administradores. O arquivo a ser gerado pelo OnCall Manager deve possuir a extensão .xls, que é o formato de planilhas do software Microsoft Excel. A figura 5.3 mostra parcialmente uma planilha com os campos anteriormente especificados preenchidos.

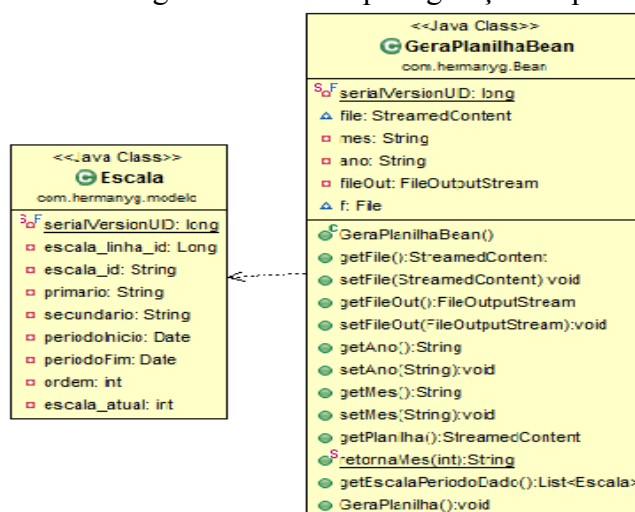
Figura 5.9 – Planilha parcial com a horas de sobre aviso do período Abril-Maio 2015

Data	Competência	Dia da Semana	Plantonista	Motivo	Horas de Plantão
20/04/2015	Abril		2 Adalberto Pereira	Plantão Suporte a Produção	16
21/04/2015	Abril		3 Adalberto Pereira	Plantão Suporte a Produção	16
22/04/2015	Abril		4 Adalberto Pereira	Plantão Suporte a Produção	16
23/04/2015	Abril		5 Adalberto Pereira	Plantão Suporte a Produção	16
24/04/2015	Abril		6 Adalberto Pereira	Plantão Suporte a Produção	16
25/04/2015	Abril		7 Adalberto Pereira	Plantão Suporte a Produção	24
26/04/2015	Abril		1 Adalberto Pereira	Plantão Suporte a Produção	24
20/04/2015	Abril		2 João Coutinho	Plantão Suporte a Produção	16
21/04/2015	Abril		3 João Coutinho	Plantão Suporte a Produção	16
22/04/2015	Abril		4 João Coutinho	Plantão Suporte a Produção	16
23/04/2015	Abril		5 João Coutinho	Plantão Suporte a Produção	16
24/04/2015	Abril		6 João Coutinho	Plantão Suporte a Produção	16
25/04/2015	Abril		7 João Coutinho	Plantão Suporte a Produção	24
26/04/2015	Abril		1 João Coutinho	Plantão Suporte a Produção	24

Fonte: Imagem criada pelo autor

Do ponto de vista das estruturas de dados envolvidos nessa funcionalidade foram criadas as classes apresentadas no diagrama de classes da figura 5.10. Essas classes possuem todos os atributos e operações necessárias para a funcionalidade de geração da planilha financeira.

Figura 5.10 – Diagrama de classe para geração da planilha financeira



Fonte: Imagem criada pelo autor

A instalação do produto com as funcionalidades descritas anteriormente com a versão de código de produção ainda não feita e por tanto o sistema ainda não está em uso e não temos dados de avaliação formais dos usuários analisando a interface e as funcionalidades de cada módulo. Levando em conta uma avaliação informal das funcionalidades apresentadas em ambiente de testes do OnCall Manager percebe-se a satisfação dos usuários com o produto entregue. A instalação da aplicação em um servidor será feita em breve, então será feito um trabalho junto aos usuários do sistema recolhendo sugestões e críticas. Os resultados dessa pesquisa de usabilidade serão analisados e se existirem propostas de melhorias que vão ao encontro das regras de negócio do OnCall Manager elas podem ser implementadas agregando valor ao produto.

6 CONCLUSÃO

Do ponto de vista de desenvolvedor a organização do desenvolvimento do produto sobre uma metodologia de desenvolvimento, no caso a metodologia ágil com base no framework scrum, foi crucial no sucesso do projeto. A visão sobre o estado do projeto após cada sprint de desenvolvimento foi importante para dimensionar a quantidade de esforço necessário para cumprir as metas e prazos de entrega do projeto. O domínio de tecnologias para a codificação de um projeto, transformando uma ideia em um produto, a organização de trabalho e tarefas disputam o papel principal no processo de desenvolvimento de software.

Ainda, as ferramentas e bibliotecas contribuem muito para aumentar a produtividade das entregas de funcionalidades do sistema fazendo com que o desenvolvedor possa concentrar esforços no desenvolvimento da lógica da funcionalidade a ser entregue.

A aplicação possui algumas limitações, por exemplo, a funcionalidade de criação de requisitos no *checklist* e a funcionalidade que cria itens na *knowledge base* não permite customizações do texto como mudar a cor da fonte por exemplo.

6.1 Trabalhos futuros

O backlog do produto ainda apresenta outras histórias que podem agregar funcionalidades futuras ao produto e alguns defeitos que precisam ser corrigidos. Como trabalhos futuros no projeto podem considerar a criação de um novo módulo no OnCall Manager que adicione funcionalidades de geração de reportes sobre os acionamentos registrados, por exemplo, reportes de quais componentes da aplicação suportada geram mais acionamentos, quantos acionamentos a equipe de suporte atendeu em um determinado período, qual o tempo médio para resolução de acionamentos.

REFERÊNCIAS

- PRESSMAN, Roger S. **Software Engineering: A practitioner's Approach**. 7th ed. Boston: McGraw-Hill Higher Education, 2010.
- JACOBSON, I. A Resounding 'Yes' to Agily Process – But Also More. **Cutter IT Journal**, [S. l.] vol. 15, n. 1, p 18-24, Janeiro 2002
- BURNS, E.; SCHALK, C. **JavaServer Faces 2.0**, The Complete Reference. 1st ed. [S. l.]:McGraw-Hill Higher Education, 2010.
- VUKOTIC, A.; GOODWILL, J. **Apache Tomcat 7**. 1st ed. [S. l.]:Apress, 2011.
- AMMANN, P.; OFFUTT, J. **Introduction to Software Testing**. Cambridge:Cambridge University Press, 2008.
- SCHWABER, K.; SUTHERLAND, J. **Um guia definitivo para o Scrum: As regras do jogo**. Julho 2013. Disponível em <<http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>>. Acesso em: 08 de jul. 2015.
- CABRAL, S. K.; MURPHY, K. **MySQL Administrator's Bible**. 1st ed. Indianapolis:Wiley Publishing, Inc, 2009
- JUNIT. **Documentação e Download Junit**. Disponível em <<https://github.com/junit-team/junit/wiki>> Acesso em: 01 de jul. 2015.
- ECLEMMA. **Documentação e Download EclEmma**. Disponível em <<http://www.eclEmma.org/>>. Acesso em: 05 de jul. 2015.
- MYSQL. **Download MySQL**. Disponível em <<http://www.mysql.com/>>. Acesso em: 05 de jul. 2015.
- HIBERNATE. **Documentação e Download Hibernate**. Disponível em <<http://hibernate.org/orm/>>. Acesso em: 25 de jun. 2015.
- TORTOISESVN. **Documentação Tortoise**. Disponível em <<http://tortoisesvn.net/about.html>>. Acesso em: 08 de jul. 2015.