

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**EVALDO ANDRADE DOS ANJOS
LEANDRO ALVES DE FONTES SILVA
VILMAR ROGOSKI JUNIOR**

**UTILIZAÇÃO DA METODOLOGIA SCRUM E FRAMEWORK JAGUAR
NO DESENVOLVIMENTO DE FERRAMENTA WEB DE APOIO A
IMPLANTAÇÃO DE PLANOS DE CARGOS E SALÁRIOS**

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA

2012

**EVALDO ANDRADE DOS ANJOS
LEANDRO ALVES DE FONTES SILVA
VILMAR ROGOSKI JUNIOR**

**UTILIZAÇÃO DA METODOLOGIA SCRUM E FRAMEWORK JAGUAR NO
DESENVOLVIMENTO DE FERRAMENTA WEB DE APOIO A IMPLANTAÇÃO DE
PLANOS DE CARGOS E SALÁRIOS**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Tecnólogo do curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Coordenação de Informática da Universidade Tecnológica Federal do Paraná.

Orientador: Prof^a MSc Simone de Almeida

PONTA GROSSA

2012



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Ponta Grossa

Diretoria de Graduação e Educação Profissional



TERMO DE APROVAÇÃO

UTILIZAÇÃO DA METODOLOGIA SCRUM E FRAMEWORK JAGUAR NO
DESENVOLVIMENTO DE FERRAMENTA WEB DE APOIO A IMPLANTAÇÃO DE
PLANOS DE CARGOS E SALÁRIOS

por

EVALDO ANDRADE DOS ANJOS
LEANDRO ALVES DE FONTES SILVA
VILMAR ROGOSKI JUNIOR

Este Trabalho de Conclusão de Curso (TCC) foi apresentado(a) em 06 de junho de 2012 como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistema. Os(as) candidatos(as) foram arguidos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Simone de Almeida
Profª. Orientadora

Cristian Cosmoski Rangel de Abreu
Membro titular

Wellton Costa de Oliveira
Membro titular

Helyane Bronoski Borges
Responsável pelos Trabalhos
de Conclusão de Curso

Simone de Almeida
Coordenadora do Curso
UTFPR - Câmpus Ponta Grossa

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

Dedicamos este Trabalho de Conclusão de Curso aos nossos familiares, professores e amigos que nos deram apoio nessa importante etapa da vida acadêmica.

AGRADECIMENTOS

Nosso agradecimento inicial é direcionado a Professora Simone de Almeida pela orientação, apoio, esforço e paciência.

Agradecemos à Professora Mônica Hoeldtke Pietruchinski pela fase inicial de orientação e a todos os professores que nos ajudaram direta e indiretamente na concepção deste trabalho.

Aos nossos familiares pelo apoio e incentivo desde o primeiro dia de aula da pré-escola até a futura graduação.

Agradecemos também aos colegas e amigos da UTFPR por sua ajuda, tanto durante o desenvolvimento deste trabalho quanto em toda a caminhada desde os primeiros períodos.

“A melhor maneira de prever o futuro é
inventá-lo” (KAY, ALAN, 1971)

RESUMO

ANJOS, Evaldo A., ROGOSKI, Vilmar Jr., SILVA, Leandro A. F. **Utilização da metodologia Scrum e framework Jaguar no desenvolvimento de ferramenta web de apoio a implantação de planos de cargos e salários**. 2012. 68 f. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas), Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2012.

O plano de cargos e salários é um conjunto de regras que estipulam como os colaboradores podem progredir na empresa com relação ao salário, porém esse processo não é trivial. Portanto, este trabalho apresenta uma alternativa a otimização dos processos de implantação dos instrumentos necessários para estabelecer uma estrutura congruente de salários, permitindo à Gestão adequar cargos à evolução da estrutura organizacional e os salários à realidade. Para o planejamento e administração do processo de criação do sistema, foram selecionados a metodologia de desenvolvimento ágil Scrum e o framework de domínio Jaguar com o objetivo de obter uma solução de código livre para otimizar as tarefas envolvidas, eliminando grande parte da codificação manual e aplicando o reuso. Como resultados, foi verificada a eficiência do framework Jaguar e a experiência em utilizá-lo juntamente à uma metodologia ágil e como esse conjunto de ferramentas pode complementar o desenvolvimento da aplicação.

Palavras-chave: Cargos e Salários. Framework Jaguar. Metodologia Ágil Scrum.

ABSTRACT

ANJOS, Evaldo A., ROGOSKI, Vilmar Jr., SILVA, Leandro A. F. **Using the Scrum methodology and the Jaguar framework to develop a web-based tool to support the implementation of plans for Jobs and wages.** 2012. 68 f. Course Conclusion Work (Analysis and Technology in System Development), Federal Technological University of Paraná. Ponta Grossa, 2012

The plan for jobs and wages is a set of rules that stipulate how employees can advance within the company when it comes to wages, though this is not trivial. Therefore, this paper presents an alternative to optimization of deployment of the necessary tools to establish a coherent structure of wages, allowing management to adapt to changing positions of the organizational structure and wages to reality. For the planning and management of the development process of the system were selected Scrum agile development methodology and framework domain Jaguar in order to get an open source solution to optimize the tasks involved, eliminating much of the coding manual and applying reuse. As a result, we verified the efficiency of the framework Jaguar and experience in using it along with an agile methodology and how this set of tools can complement the development of the application.

Keywords: Jobs and Wages. Jaguar Framework. Scrum Agile methodology.

LISTA DE FIGURAS

Figura 1 – Carreira com ascensão na diagonal	22
Figura 2 – Descritivo do processo de implantação do PCS.....	23
Figura 3 – Diagrama de camadas da arquitetura Jaguar	30
Figura 4 – Arquitetura da IDE de desenvolvimento	31
Figura 5 – Ciclo de trabalho do Scrum	34
Figura 6 – Diagrama de blocos	39
Figura 7 – Painel de administração do Scrumhalf	42
Figura 8 – Interface do Product Backlog	43
Figura 9 – Use case Manter Setor	44
Figura 10 – Quadro de tarefas	45
Figura 11 – Tela para novo projeto no Jaguar	47
Figura 12 – Assistente de criação de projeto do Jaguar	48
Figura 13 – Inicialização do Apache Derby	48
Figura 14 – Criação da classe Setor	49
Figura 15 – Mapeamento Objeto-Relacional.....	50
Figura 16 – Código da classe Setor	52
Figura 17 – Assistente de criação da visão	53
Figura 18 – Opções da página	54
Figura 19 – Assistente de manutenção da página	55
Figura 20 – Prévia do cadastro de setor	55
Figura 21 – Tela de login	56
Figura 22 – Página inicial da administração	56
Figura 23 – Gerador de esquema de banco de dados	57
Figura 24 – Tela de cadastro de setor	58
Figura 25 – Tela de login	59
Figura 26 – Tela inicial	60
Figura 27 – Tela de manutenção de funcionários	60
Figura 28 – Tela manutenção de UF's	61
Figura 29 – Tela manutenção de setores	62
Figura 30 – Tela manutenção de municípios	62
Figura 31 – Tela manutenção de cargos	63

LISTA DE TABELAS

Tabela 1 – Escala de pontos para avaliação dos cargos	26
Tabela 2 – Divisão de carreiras	26
Tabela 3 – Exemplo de graus e níveis	27

LISTA DE QUADROS

Quadro 1 – Etapas da implantação do plano de cargos e salários	19
Quadro 2 – Exemplo de quadro funcional	25

LISTA DE SIGLAS

API	<i>Application programming interface – Aplicação de programação de interface</i>
IDE	<i>Integrated Development Environment – Ambiente integrado de desenvolvimento</i>
JEE	<i>Java Enterprise Edition – Edição Empresarial Java</i>
JVM	<i>Java Virtual Machine – Máquina Virtual Java</i>
SQL	<i>Structured Query Language – Linguagem de Consulta Estruturada</i>
XML	<i>extensible Markup Language – Linguagem Extensiva de Marcação</i>
GPL	<i>General Public License – Licença Pública Geral</i>
CRUD	<i>Create, Read, Update, Delete – Criar, Ler, Atualizar, Excluir</i>
MVC	<i>Model-view-controller – Modelo, Visão, Controle</i>
JPA	<i>Java Persistence API – Aplicação de Persistência Java</i>
DDL	<i>Data Definition Language – Linguagem de Definição de Dados</i>
URL	<i>Uniform Resource Locator – Localizador de Recursos Uniformes</i>
RH	Recursos Humanos

SUMÁRIO

1 INTRODUÇÃO	15
1.1 OBJETIVO GERAL	17
1.2 OBJETIVOS ESPECÍFICOS	17
1.3 ESTRUTURA DO TRABALHO	18
2 REFERENCIAL TEÓRICO	19
2.1 PLANOS DE CARGOS E SALÁRIOS	19
2.1.2 Métodos	23
2.2 JAGUAR	28
2.2.1 História.....	28
2.2.2 Conceitos.....	28
2.2.3 Vantagens	29
2.2.4 Estrutura e Funcionamento	29
2.3 CONSIDERAÇÕES FINAIS DO CAPÍTULO	32
3 METODOLOGIA SCRUM	33
3.1 HISTÓRIA	33
3.2 CONCEITO.....	33
3.2.1 Papéis	34
3.2.2 Cerimônias	36
3.2.3 Artefatos	36
3.3 CONSIDERAÇÕES SOBRE O CAPÍTULO	37
4 PROTÓTIPO	38
4.1 DESCRIÇÃO	38
4.2 O <i>FRAMEWORK</i> JAGUAR NO DESENVOLVIMENTO	40
4.3 A METODOLOGIA SCRUM NO DESENVOLVIMENTO	41
4.4 CONSIDERAÇÕES SOBRE O CAPÍTULO	45
5 DESENVOLVIMENTO DA APLICAÇÃO	46
5.1 ETAPAS DO DESENVOLVIMENTO.....	46
5.2 RESULTADOS.....	58
5.2 CONSIDERAÇÕES SOBRE O DESENVOLVIMENTO DA APLICAÇÃO	63
6 CONCLUSÃO E TRABALHOS FUTUROS	65
6.1 CONCLUSÃO	65

6.2 TRABALHOS FUTUROS	66
REFERÊNCIAS	67

1 INTRODUÇÃO

Com o processo de globalização, surgimento de empresas multinacionais, aumento da competitividade e com o desenvolvimento tecnológico, a velocidade da informação no ramo empresarial teve um grande aumento, tanto no administrativo quanto nos recursos humanos. A gestão de pessoas contrasta com ideais antigos como a Administração Científica de Frederick Taylor que via o processo como mais importante que o funcionário, considerado como mera ferramenta de trabalho.

Do antigo departamento pessoal administrador de cargo, salário, função, taxas e aposentadoria, passa-se para a ideia de recursos humanos, que possui uma atitude mais humanizada com os funcionários, não sendo interpretados como máquinas em uma linha de montagem.

Do conceito de Recursos Humanos, surgiu a ideia da Gestão de Pessoas, que é caracterizada pela participação, capacitação, envolvimento e desenvolvimento do capital humano que são as pessoas que a compõem. Cabe a área de gestão de pessoas a função de humanizar as empresas. (MIRANDA, MIRANDA, 2009).

Entre os interesses debatidos nesse modelo de gestão, está a Administração de Cargos e Salários que visa mensurar o valor que cada função deve receber como retribuição por sua execução. Mas apenas estipular um valor para cada cargo e manter este por um período muito longo de tempo, além de desatualizar essa remuneração em termos econômicos, seja pela inflação ou pela desvalorização da moeda, também tira do funcionário um horizonte de crescimento pessoal e profissional. Para criar este horizonte é preciso contar com um ambiente que valorize a motivação dos funcionários.

Dentre as soluções para criar uma estratégia salarial que concilie uma remuneração justa acompanhada de um plano de recompensas que define um horizonte de crescimento para o funcionário dentro daqueles que foram definidos como seu cargo e a qual nível deste cargo ele pertence, surgiu a elaboração do Plano de Cargos e Salários.

Um Plano de Cargos e Salários tem a função de criar uma estrutura de remuneração, que é segundo Tachizawa et al.(2006, p. 180) “o conjunto de vantagens que uma pessoa recebe em contrapartida pela prestação de um serviço”.

Atividades como a elaboração de um Plano de Cargos envolvem uma grande quantidade de pesquisas, materiais e questionários, gerando muitas

informações, das quais na maioria das vezes são de exclusividade e confidencialidade da empresa em que o projeto esteja sendo implementado. A informatização destes processos surge como uma alternativa, seja para a coleta de dados ou para a agilidade na transmissão das informações e sua segurança. O processo de informatização, pode ser ainda mais otimizado com a existência de metodologias para guiar um desenvolvimento por meio de padrões que são responsáveis por traçar uma estratégia de desenvolvimento desde as reuniões iniciais com o cliente até a entrega do produto final bem como seu posterior suporte.

Metodologias de desenvolvimento Ágil têm como característica a existência de uma liderança, responsável por motivar a equipe, concentrá-la nos resultados e principalmente atender as necessidades do cliente.

Uma dessas metodologias é chamada *Scrum*, e visa um método de gerenciamento de projetos simplificado através do desenvolvimento iterativo, ou seja, por meio de ciclos de repetição e incremental, onde a cada ciclo o conteúdo do projeto pode ser adaptado, melhorado e ampliado. Segundo (PRESSMAN, 2006, p. 720):

O *Scrum* pode tornar mais dinâmico e gerenciável em um ambiente de desenvolvimento de sistemas, pois faz uso de conceitos herdados do *RUP*, os quais visam unir equipes em benefício de um objetivo comum. Essa metodologia tem como base a divisão do problema em partes menores para facilitar o desenvolvimento.

Através de um processo denominado *ProductBacklog*, estas partes menores são definidas pela prioridade do cliente e tem um período de tempo pré determinado para serem desenvolvidas, cada um desses períodos de tempo é conhecido como *Sprint* e normalmente tem a duração de uma semana a um mês. O Cliente poderá ter uma demonstração de cada parte do produto ao final de cada *Sprint*. Toda essa cadeia de métodos é repetida para todos os *ProductBacklog's* até que o desenvolvimento esteja terminado.

Para complementar a tarefa de desenvolvimento, com ou sem uma metodologia pré-definida, pode ser utilizado um recurso chamado *Framework* ou arcabouço. Segundo Fayad (1997), *frameworks* representam uma estrutura formada por blocos pré-fabricados de *software* que os programadores podem utilizar, estender ou adaptar para uma solução específica e linguagens de padrões.

O Jaguar foi lançado em novembro de 2010 na Conferência Latino Americana de *Software* Livre, Latinoware. É baseado no *framework* JCompany, e entregue a comunidade de *Software* Livre como uma alternativa, principalmente para o desenvolvimento de ferramentas para o setor público, mas que também pode ser utilizado pela iniciativa privada. Trabalha numa arquitetura de alto nível possibilitando reutilização e extensão. Para sua criação vários *frameworks* de código livre foram utilizados.

Este trabalho apresenta o desenvolvimento de um sistema web para administração de Planos de Cargos e Salários e Avaliação de Desempenho, utilizando-se do *Framework* Jaguar e da metodologia *Scrum na sua construção*.

1.1 OBJETIVO GERAL

Desenvolver um sistema web de auxílio à implantação de Planos de Cargos e Salários, demonstrando como o desenvolvimento ágil pode ser útil na criação de um sistema através da metodologia *Scrum*, bem como denotar os benefícios da padronização ao adotar-se um *framework*, aqui representado pela escolha do Jaguar.

1.2 OBJETIVOS ESPECÍFICOS

- Identificar as características, diretrizes e restrições do processo de elaboração e administração do Plano de Cargos e Salários;
- Compreender a utilização do *framework* Jaguar no desenvolvimento dos aplicativos;
- Aplicar as tecnologias estudadas no desenvolvimento de um sistema de Plano de Cargos e Salários;

1.3 ESTRUTURA DO TRABALHO

O estudo sobre a aplicação do *framework* Jaguar no problema de gestão de plano de cargos e salário e avaliação de desempenho foi organizado em 6 capítulos, sendo este o primeiro.

O capítulo 2 faz uma revisão dos princípios da elaboração de um plano de cargos e salários e descreve a estrutura e funcionalidades do *framework* Jaguar e sua aplicabilidade no projeto proposto.

A descrição da metodologia *Scrum* e como foi utilizada no desenvolvimento do projeto é referida no capítulo 3.

No capítulo 4 é apresentada a aplicação dos conceitos referentes ao *framework* Jaguar em conjunto com a metodologia de desenvolvimento Scrum para desenvolvimento da aplicação de apoio ao gerenciamento de cargos e salários.

No Capítulo 5 é apresentada a implementação do módulo setor da aplicação demonstrada através dos passos seguidos para desenvolvê-la com o *framework* Jaguar, bem como os resultados obtidos no trabalho com as ferramentas utilizadas e com o produto final.

O capítulo 6 finaliza este trabalho, apresentando as conclusões, bem como possíveis trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo tem como objetivo apresentar o conceito e as metodologias para elaborar um plano de cargos e salários e como a informatização pode contribuir para auxiliar os processos envolvidos. Objetiva também a descrição do *framework* Jaguar, sua origem, particularidades e processos.

2.1 PLANOS DE CARGOS E SALÁRIOS

Um plano de cargos e salários é uma estratégia de remuneração pertencente à área de recursos humanos, conhecida como Administração de salários, sua importância no entendimento de (PONTES, 2005, p.31)

Para que possa existir ambiente motivador no seio da organização, pessoas integradas e produtivas, são necessários planos adequados de Gestão de Recursos Humanos. O plano básico é, sem dúvida, o de cargos e salários, porque sem ele dificilmente os demais planos de desenvolvimento funcionam.

A implantação do plano depende de diversas etapas e de uma colaboração consistente entre funcionários, gerência e o pessoal que faz a elaboração dos planos, que pode ser o próprio setor de RH da empresa em questão ou uma empresa externa de consultoria.

O Quadro 1, descreve os passos necessários para a elaboração de um plano de cargos e salários consistente segundo (PONTES, 2005):

Quadro 1 – Etapas da implantação do plano de cargos e salários

Etapas	Fases Importantes
Planejamento e Divulgação do plano	Elaboração do plano Discussão do plano com as gerências Aprovação do plano Divulgação do plano aos colaboradores
Análise dos cargos	Coleta de dados Descrição dos cargos Especificação dos cargos Titulação dos cargos Classificação dos cargos conforme os grupos ocupacionais Catálogo de cargos
Avaliação dos cargos	Escolha de cargos chave Formação do Comitê de avaliação

	Escolha do método de avaliação Avaliação dos cargos chave Avaliação dos demais cargos
Pesquisa salarial	Escolha dos cargos Escolha das empresas Elaboração do manual de coleta de dados Coleta de dados Tabulação de dados Análise de resultados Relato às empresas participantes
Estrutura salarial	Cálculo da curva básica Determinação do número de graus Cálculo das amplitudes dos graus Análise de sobreposição dos graus
Política salarial	Salário de admissão Promoção horizontal e vertical Reavaliação dos cargos Reajustes salariais Nível hierárquico das aprovações
Política de remuneração	Definição de adicionais salariais Definição dos pacotes de benefícios Composição da remuneração
Carreiras profissionais	Sucessão de cargos Divulgação das carreiras aos colaboradores Definição do planejamento de recursos humanos Programa de desempenho Programa de treinamento e desenvolvimento
Participação nos lucros ou resultados	Definição da estrutura do programa

Fonte: Pontes (2005, p. 36)

A síntese das etapas propostas por Pontes (2005) é a ideia de que o início dos trabalhos pré-implementação dos planos deve se dar por uma definição objetiva dos objetivos a serem atingidos com a implantação. Ainda segundo Pontes (2005) esses objetivos são:

- a) Atender a premissa do equilíbrio interno e externo;
- b) Servir como instrumento das oportunidades de trabalho e desenvolvimento na empresa;
- c) Tornar clara a política de salários;
- d) Servir como base de um plano de carreiras e de outros subsistemas de recursos humanos, como treinamento e desenvolvimento, planejamento, recrutamento e seleção;
- e) Provocar nas pessoas motivação e estímulo para melhorar suas qualificações profissionais.

A premissa de equilíbrio interno e externo é dada pela definição de consistência, onde deve haver igualdade de salários se comparadas ao mercado chamadas de consistência externa, ou seja, realizando um paralelo entre os salários aplicados na

empresa que pretende implementar o plano de cargos e salários e das outras empresas da sua região ou ramo de atividade. Também existe a consistência interna onde são classificados os cargos internos a empresa.

Antes de trabalhar diretamente com as estratégias salariais, ainda no processo de busca pela consistência interna, é necessário realizar a avaliação dos cargos da empresa.

O processo de analisar e comparar o conteúdo dos cargos, no sentido de colocá-los em uma ordem de classes que proporcione a estrutura do sistema de remuneração. Trata-se de um meio para se determinar o valor relativo de cada cargo dentro de uma estrutura organizacional, isto é, relativa de cada cargo diante dos demais (CHIAVENATO, 2005, p. 284).

A definição da consistência interna é feita após serem traçados os objetivos do plano de cargos e salários o que depende de número dos chamados grupos ocupacionais que são os tipos de cargos encontrados na mesma empresa, sejam eles operacionais, administrativos, técnicos ou gerenciais. Com a definição dos cargos devem ser discutidas e estabelecidas às metodologias para a sua implantação de acordo com o ramo de atividade da empresa e da quantidade de funcionários, também deve ser levada em conta a divisão em etapas para a implantação e os custos que ela irá ter, depois de uma síntese de todos esses fatores a implantação deve ser discutida com a gerência da empresa para que haja ou não o aceite por parte da chefia, caso sim, o processo deverá ser amplamente divulgado para os demais funcionários.

Outro detalhe importante é definir como será feita a definição das chamadas promoções verticais e horizontais, onde promoção horizontal é:

Na passagem do colaborador para uma classe superior de salário no mesmo cargo e grau decorrente de destacado desempenho de suas tarefas e aumento da experiência. Em geral, a promoção horizontal obedece ao prazo mínimo de um ano entre um aumento e outro (PONTES, 2005, p. 303).

e promoção vertical é:

Passagem do colaborador de um cargo para outro, classificado em grau superior. Em geral a promoção vertical obedece ao prazo mínimo de seis meses de permanência do funcionário no cargo inicial (PONTES, 2005, p. 303).

Já o autor Bergue (2005, p. 220), propõe uma carreira em ascensão diagonal como mostra a Figura 1:

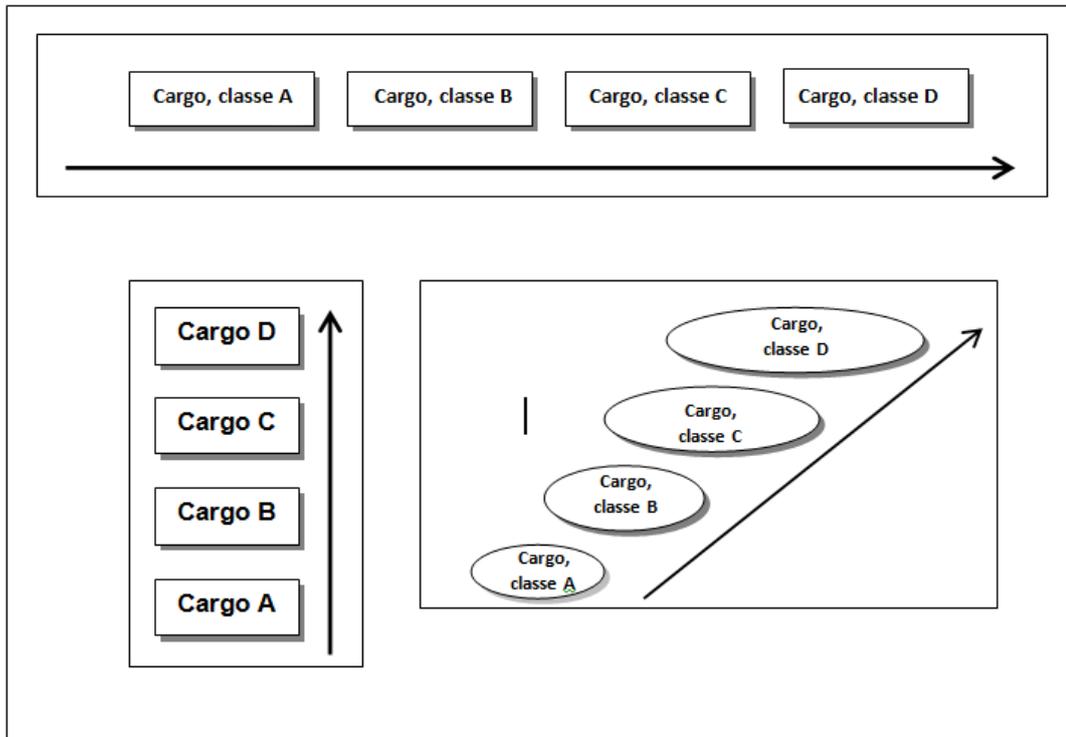


Figura 1 – Carreira com ascensão na diagonal
Fonte: Bergue (2005, p. 220)

Na Figura 1 Bergue (2005, p. 220), cita que “o aumento das elipses que representam os cargos crescendo na diagonal como a transformação das competências de um cargo durante sua carreira”.

Para que haja tal ascensão deve haver desafios profissionais e pessoais que estejam diretamente ligados a essa trajetória de ascensão, que também deverá ser interdependente da curva de remuneração do funcionário, relacionando a eficiência funcional ao seu desempenho no cargo. Essa relação deve seguir normas que segundo Bergue (2005, p. 221) são:

- a) Critérios de ascensão nas carreiras;
- b) Conjunto de exigências para cada nível da carreira;
- c) Nível de remuneração de cada classe na carreira;
- d) Critérios e procedimentos para enquadramento;
- e) Política institucional de formação e desenvolvimento de competências valorizadas pela organização.

2.1.2 Métodos



Figura 2 – Descritivo do processo de implantação do PCS
Fonte: Autoria Própria

A pesquisa envolvida no desenvolvimento de um plano de cargos e salários, define a escolha de um ou mais métodos para auxiliar na criação do plano, são métodos que podem ser quantitativos, quando envolvem contagem de funções, cargos, salários e funcionários e qualitativos, quando comparam e descrevem características.

O início da pesquisa dá-se com a definição da chamada população alvo, ou seja, quantos funcionários a empresa tem. Segundo Gil (2008, p.97) “em organizações menores o levantamento de informações pode ser feito com a totalidade de empregados, já em empresas maiores, pode ser feita com um plano de amostragem”.

A pesquisa quantitativa e parte da pesquisa qualitativa podem ser auxiliadas pela elaboração de um questionário, a ser respondido pelos funcionários da empresa com dados sobre o cargo, idade, escolaridade, tempo de empresa e caso julgue-se necessário, questões sobre o que o funcionário acha sobre a implantação do plano de cargos e salários.

Após a pesquisa é preciso subdividir as categorias de funcionários de acordo com a estrutura organizacional proposta pela empresa. Em alguns casos a estrutura pode ser reorganizada e subdividida em famílias, normalmente organizada pela formação e atividades desempenhadas. Abaixo, um exemplo de descrição de cargo, a partir dele, o funcionário pode ser alocado na família ao qual pertence na estrutura organizacional:

O levantamento dos cargos foi realizado através do preenchimento de formulários no modelo exemplificado:

Título do cargo: Auxiliar administrativo.

Descrição sumária: digitar, redigir, arquivar e receber documentos, atender pessoas.

Descrição geral:

- a) Atendimento aos demais funcionários;
- b) Utilização de equipamento de informática e programas de escritório, impressões e arquivamento de documentos;
- c) Redação de correspondência;
- d) Atendimento ao público em geral;

Requisitos:

- a) Nível de escolaridade: Ensino médio concluído;
- b) Nível de especialização: Não exigido;
- c) Experiência exigida: Seis meses em ocupação semelhante.
- d) Aptidões necessárias adicionais: boa digitação, boa redação, responsabilidade, pontualidade, boa apresentação, bom relacionamento interpessoal.

Condições de trabalho:

Predominantemente no escritório com poucas atividades no ambiente de produção. O Quadro 2 é um exemplo de organização de um quadro funcional:

Quadro 2 – Exemplo de quadro funcional

Família Ocupacional	Atividades	Formação
Administrativa	Auxiliar, assistente administrativo	Ensino médio, sem formação especializada.
Técnica Operacional	Auxiliar técnico	Ensino fundamental sem formação especializada.
Técnica de nível médio	Técnico especializado	Ensino médio técnico/profissionalizante.
Nível Superior	Especialista	Ensino superior específico

Fonte: Autoria própria

Para realizar a avaliação de cargos existem diversos métodos descritivos, dentre os mais utilizados, vale citar o método do escalonamento, também conhecido como comparação simples ou *job ranking*, o método das categorias predeterminadas ou escalonamentos simultâneos, o método da comparação de fatores e o método da avaliação por pontos.

O método do escalonamento consiste em listar os cargos por nível de importância de acordo com critérios pré-determinados para compará-los. Trata-se de comparar os cargos um a um procurando situar no topo os que tradicionalmente são considerados mais importantes (TACHIZAWA et al., 2006).

O Método das categorias predeterminadas tem como base o método do escalonamento, porém, os cargos são separados inicialmente em conjuntos que possuam características em comum, onde para cada categoria haverá um cargo de referência para fatores comparativos.

Dentre os métodos também é apresentado o Método da comparação de fatores, que define fatores a serem levados em consideração de acordo com uma análise prévia dos cargos. Nessa análise são levados em consideração os fatores chamados de requisitos mentais como escolaridade, experiência de trabalho, iniciativa e demais fatores voltados ao ponto de vista intelectual do indivíduo. As chamadas habilidades requeridas também são levantadas na análise, num paralelo traçado entre as exigências de cada cargo como habilidades mentais, manuais e concentração.

Os requisitos físicos também precisam ser levantados de acordo com o esforço e condição física requerida pelo cargo em questão. O fator responsabilidade

é levado em consideração no sentido da importância que o cargo tem para a empresa no âmbito de processos, equipamentos, capital e informação. Por último são analisadas as condições de trabalho levando em conta o ambiente físico e riscos envolvidos em cada cargo.

O método de avaliação por pontos é o mais utilizado e aperfeiçoado (CHIAVENATO, 2005). Nele os cargos são comparados de acordo com graus de avaliação, definidos por fatores como no método da comparação.

Os cargos são comparados em relação a cada fator de avaliação e o resultado é transformado em pontos, visando avaliação quantitativa. Essa comparação é feita por meio de um formulário denominado matriz de avaliação de cargos (CHIAVENATO, 2005, p. 287).

Somando-se os pontos de avaliação em cada um dos fatores de avaliação da matriz de avaliação de cargos obtém-se a avaliação final de cada cargo. A partir daí cada cargo é identificado pelo seu valor em pontos. Os valores em pontos dos diversos cargos são transferidos para um gráfico cartesiano, em que o eixo das coordenadas (y) constitui os salários e o eixo das abscissas (x) os valores em pontos. Com essa distribuição dos valores de salários e de pontos, traça-se a linha de tendência que constituirá a curva salarial resultante.

Fatores de avaliação de cargos	Grau A	Grau B	Grau C	Grau D	Grau E
Requisitos mentais:					
1. Escolaridade	20	40	60	80	100
2. Experiência	20	40	60	80	100
3. Iniciativa	15	30	45	60	75
Requisitos físicos:					
4. Esforço físico	10	20	30	40	50
Responsabilidades:					
5. Supervisão	20	40	60	80	100
6. Equipamentos	5	10	15	20	25
Condições de trabalho:					
7. Ambiente	10	20	30	40	50

Tabela 1 – Escala de pontos para avaliação dos cargos.

Fonte: (Chiavenato 2005, p. 288)

Depois de realizada a subdivisão dos cargos e das famílias será possível a criação de uma tabela como a do exemplo abaixo:

Carreira	Número de Funcionários	Porcentagem
Administrativa	40	20%
Técnica Operacional	60	30%

Técnica de nível médio	80	40%
Nível Superior	20	10%
Totais	200	100%

Tabela 2 – Divisão de carreiras
Fonte: Autoria própria

Com uma definição objetiva dos cargos e carreiras é possível retornar ao conceito de evolução vertical e horizontal para a posterior definição dos salários. Na evolução horizontal o funcionário permanece no mesmo cargo, porém subindo de grau, estes, têm seu critério estabelecido pela empresa contratante, bem como os requisitos para passar de um grau ao próximo, sendo normalmente utilizados tempo de empresa ou desempenho funcional. Na Tabela 3 um exemplo de evolução horizontal para um técnico de nível médio onde é necessário passar por cinco graus antes de ascender na classificação.

Nível	Junior					Pleno					Sênior					Master				
Grau	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5

Tabela 3 – Exemplo de graus e níveis
Fonte: Autoria própria

A ascensão vertical ocorre quando o funcionário muda de cargo e pode ser determinada no plano de cargos e salários para que se tenha uma estrutura organizacional mas justa, como por exemplo impedir que um funcionário de nível Júnior venha a ser chefe de um funcionário de categoria superior como Pleno ou Sênior.

Na esfera pública também é comum a ascensão de carreira ocorrendo inclusive automaticamente quando a formação acadêmica é atingida, como por exemplo, um técnico de nível médio numa área X que conseguindo se formar no nível superior nesta área é promovido a um cargo especialista.

Com as carreiras, cargos e funções bem definidos, a empresa pode tratar da estratégia salarial, normalmente realizada através de uma pesquisa em empresas do mesmo ramo de atividade ou com, pelo menos, cargos parecidos.

A palavra final em relação a todos os processos fica sempre a cargo da diretoria da empresa, após aprovação e implementação dos planos fica a cargo de todos os membros da empresa respeitá-lo e fiscalizar a sua utilização para que os procedimentos de contratação e promoção não fujam ao que foi estabelecido pelo plano.

2.2 JAGUAR

Esta seção apresenta um breve histórico do surgimento e utilização de *frameworks*, assim como alguns conceitos, vantagens de sua utilização e a estrutura e seu funcionamento.

2.2.1 História

Recentemente o Governo Federal tem demonstrado grande apoio ao uso de *Softwares Livres*, tanto pela questão financeira, onde eliminam custo da compra do *software*, quanto nas questões de segurança, que adotando-se uma política de *software* e conservando seu desenvolvimento nos próprios órgãos públicos de tecnologia da informação, dados confidenciais de grande importância para a população, não ficarão sob controle de empresas privadas. Outro ponto que contribuiu para a adoção do *software* livre no setor público foi para incentivar desenvolvedores nacionais a criar ferramentas livres e contribuir para a melhoria de ferramentas já existentes. Resultado disso foi a publicação no Diário Oficial pela Secretaria de Logística de TI (SLTI) da Instrução Normativa nº 1/2011, que regulamenta o que é um *Software Livre Brasileiro*, ou seja, uma regulamentação para a escolha de ferramentas livres que poderiam ser utilizadas como *software* na esfera da administração pública.

Seguindo a ideia do *Software Público Brasileiro*, a empresa *Powerlogic*, disponibilizou o *framework* Jaguar como uma ferramenta para auxílio no desenvolvimento de aplicações para o setor público, de código aberto, podendo ser utilizada também para aplicações de empresas privadas. O lançamento foi realizado no dia 10 de novembro de 2010 na conferência Latino Americana de *Software Livre*, *Latinoware*.

2.2.2 Conceitos

O Jaguar possui uma arquitetura de *software* de alto nível, reutilizável e extensível baseada na integração de dezenas de *frameworks* de base, para atender requisitos não funcionais importantes e universais como padrões de formulários, botões, layouts , validações e ciclo de vida de objetos.

Disponível para ambiente Eclipse com diversos *plugins* homologados, o *framework* Jaguar tem o foco na arquitetura, onde desenvolvedor somente se preocupa com a regra de negócio, tendo em vista que o *framework* Jaguar oferece uma arquitetura pronta para desenvolvimento, poupando o tempo na integração de *frameworks* de baixo nível, configuração e atualização.

2.2.3 Vantagens

Dentre as vantagens do Jaguar, pode-se destacar a padronização, permitindo que uma aplicação seja padronizada, onde usuários com diferentes níveis de conhecimento consigam manter um padrão de desenvolvimento, reduzindo assim a curva de aprendizado, pois ao invés de dominar todos os *frameworks* de base e todos os detalhes de integração dos mesmos, o usuário apenas deve se preocupar com os padrões propostos pelo Jaguar, assim uma melhor produtividade no desenvolvimento irá acontecer. É importante destacar que sua arquitetura, desenvolvida há oito anos, é constantemente atualizada com novas técnicas utilizadas no mercado de trabalho. Assim, a qualidade e mitigação de riscos não ficam por conta do desenvolvedor e sim do próprio *framework*.

O Jaguar é baseado no jCompanyDeveloper, um *framework* já estabelecido no ambiente computacional, o *framework* Jaguar é um aplicativo para desenvolvimentos de aplicações Java EE 6 em licença GPLv2 e GPLv3 (POWERLOGIC, 2012).

Segundo a Powerlogic, empresa criadora do Jaguar, a opção pelo licenciamento *Open Source* 2.0 foi importante para o projeto pois com o código aberto é possível realizar uma terceirização gerenciável da complexidade do *framework* e ainda assim gerar benefícios para a comunidade do *software* livre.

2.2.4 Estrutura e Funcionamento

Dentre as estruturas identificadas no *framework* Jaguar estão as características citadas por seu desenvolvedor como a Automação completa, onde os chamados robôs eliminam o trabalho humano em várias tarefas, eliminando a necessidade de codificação manual de partes de programas (POWERLOGIC, 2012).

A IDE do Jaguar em si, atua em áreas que não podem ser totalmente automatizadas, acelerando a codificação manual dos programas. Através do módulo chamado Patterns&Methods, o Jaguar auxilia o desenvolvedor com ferramentas de padrões de solução para problemas frequentes, documentação extensiva e tutoriais. Através do módulo *Configuration Management* é estabelecido um ambiente para monitoria dos processos visando assegurar sua estabilidade e otimização.

O Jaguar diferencia-se dos *frameworks* mais comuns por conta de sua integração que atua como um *framework de frameworks* (Powerlogic). Essa ideia parte do princípio de estímulo do reuso, utilizando os *frameworks* de base para tarefas mais comuns e se concentrando em tarefas de nível mais alto.

A Figura 2 representa o diagrama de camadas da arquitetura proposta pelo Jaguar, representado nas camadas de número 2, 3 e 4 localizados do lado esquerdo da figura.

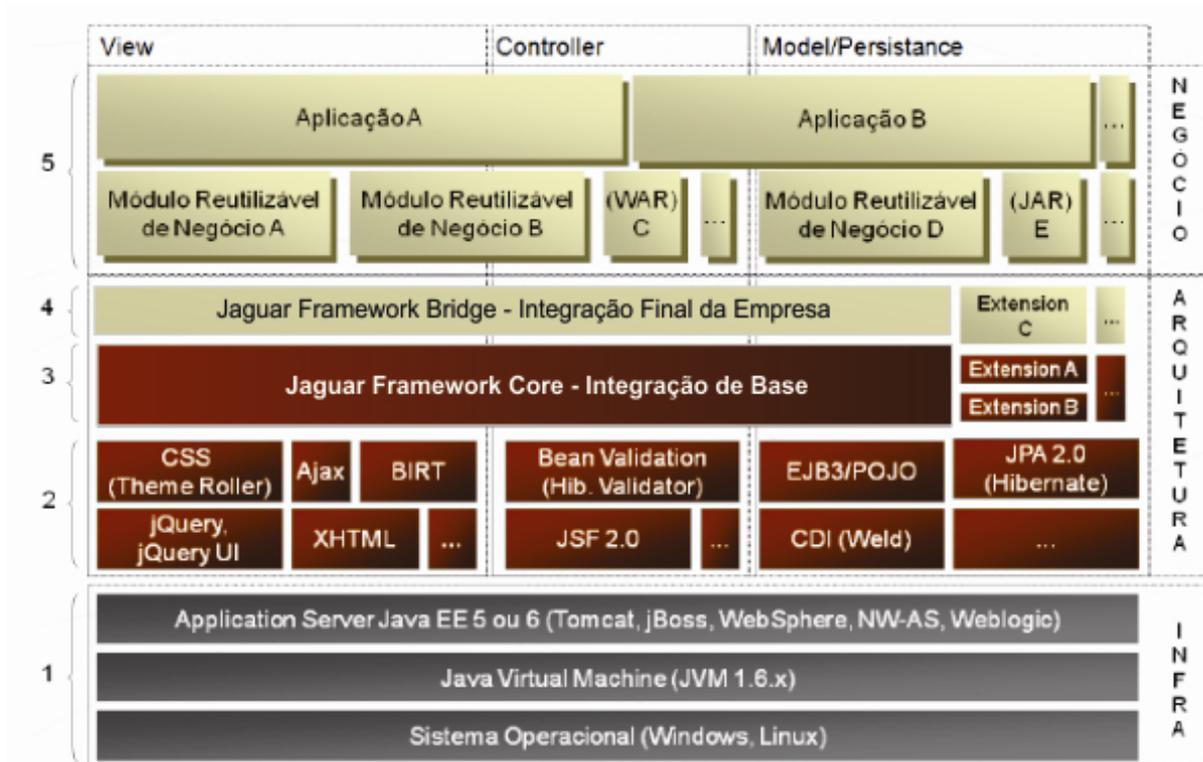


Figura 3 – Diagrama de Camadas da Arquitetura Jaguar
Fonte: (Powerlogic, 2012, p.10)

O nível um trata da camada de Infraestrutura de *software*, sob domínio da equipe de produção. O nível dois traz os *frameworks* e utilitários de base, todos *Open Source* para favorecer o reuso e utilização nas tarefas mais simples.

O nível três é a camada chamada *Core* do Jaguar. Integra os insumos da camada dois para realizar tarefas de nível mais alto.

O Nível quatro chamado de *bridge* é um *framework* de isolamento da camada inferior que permite customização e generalização das aplicações de acordo com o projeto a ser desenvolvido. Chamada de *Core Business*, a camada do nível cinco traz os módulos de negócio reutilizáveis e aplicações de padronização de arquitetura.

A IDE Jaguar (ambiente integrado de desenvolvimento), assim como o *framework* trabalha com a filosofia de reuso e especialização, utilizando também os utilitários *Open Source* em tempo de desenvolvimento como na Figura 3.

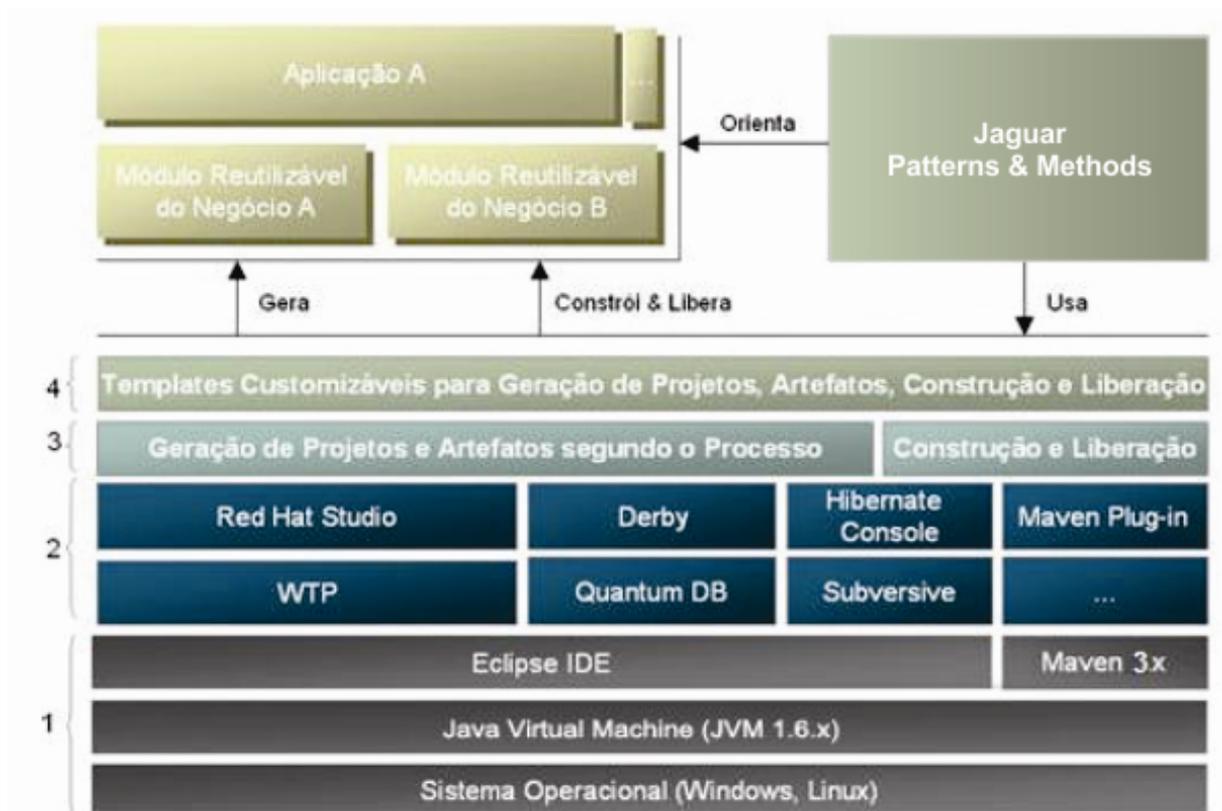


Figura 4 – Arquitetura da IDE de desenvolvimento
Fonte: (Powerlogic, 2012)

A camada denotada pelo número um trata-se da Infraestrutura de *software*, que integra o Jaguar a IDE Eclipse.

A camada representada pelo número dois apresenta o reuso da matéria prima *Open Source*, onde estão os *softwares* de base e *plug-ins* da IDE Eclipse, são recursos de apoio genérico, neutros em relação à arquitetura, padrões e métodos do Jaguar (POWERLOGIC, 2012).

A camada de número três, demonstra a camada composta por *plugins* da IDE e utilitários Maven do Eclipse que interagem com o Jaguar, bem como os padrões e métodos do *framework*.

A quarta camada responde pelas tarefas de customização, onde se encontra a camada de *templates* que respeitam as especificações passadas por meio da camada *bridge* do *framework*.

2.3 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Após o estudo realizado para compor o capítulo, foi concluído pela equipe que a ferramenta a ser desenvolvida deverá atender às tarefas de levantamento e cadastro de informações para a geração de relatórios sobre todo o processo, ficando assim, independente de um método específico, já que, as informações a serem cadastradas na ferramenta são comuns a todos os métodos de levantamento de dados para elaboração de planos de cargos e salários.

E com o estudo do *framework* Jaguar, foi caracterizada a importância de sua utilização e a viabilidade de uso no trabalho aqui referido, pois a demanda de uma ferramenta para nortear o desenvolvimento foi suprida e eliminou tarefas básicas de programação, guiando um desenvolvimento mais lógico e mais focado no resultado, deixando tarefas simples de codificação por conta do *framework*.

3 METODOLOGIA SCRUM

Este capítulo objetiva a apresentação da metodologia de desenvolvimento ágil denominada *Scrum*, bem como sua história, conceitos, procedimentos e especificidades.

3.1 HISTÓRIA

A premissa que envolve o Scrum surgiu em 1986 num artigo escrito por (HirotakaTakeuchi e IkujiroNonaka, 1986) onde foi proposto um esquema de gerenciamento de projetos para indústrias do ramo automotivo, dando importância a grupos menores e multidisciplinares envolvidos na obtenção de resultados.

A metodologia foi criada posteriormente por Jeff Sutherland e John Scuminiotales, utilizando dos métodos propostos por Nonaka e Takeuchi para implementar a metodologia, até então sem nome na empresa *Easel Corporation*.

O nome Scrum, derivado de uma jogada do esporte rúgbi onde todo o time avança como uma unidade, trabalhando em conjunto, passando a bola de um para outro, surgiu em 1995 quando Ken Schwaber apresentou a metodologia na OOPSLA (*Object-Oriented Programming, Systems, Languages & Applications*) uma conferência Norte Americana realizada anualmente desde 1986 para debater tópicos relacionados a desenvolvimento de sistemas orientados a objetos, o que ajudou o Scrum a ficar conhecido mundialmente pela comunidade de desenvolvedores (CISNEIROS, 2009).

3.2 CONCEITO

Segundo Cisneiros (2009) o Scrum é um modelo de desenvolvimento ágil de *software* que fornece métodos para se definir o planejamento, os principais papéis de pessoas e a forma de trabalho do time.

Para Schwaber (2011) o Scrum é uma estrutura processual para desenvolvimento e manutenção de produtos complexos e é apoiado em três pilares, a transparência, a inspeção e a adaptação.

A transparência é o conceito de que todos os processos envolvidos no desenvolvimento com Scrum são visíveis a todos os usuários para que a compreensão seja compartilhada entre eles.

A inspeção é o princípio em que os usuários do Scrum verificam os artefatos para detectar variações do projeto original ou detalhes que ficaram alheios ao que foi planejado.

A adaptação é a forma encontrada para voltar o projeto ao seu foco principal quando ele foi desviado de seu trajeto pré-estabelecido seja deliberadamente ou por falhas no desenvolvimento, a necessidade de adaptação é definida durante as reuniões.

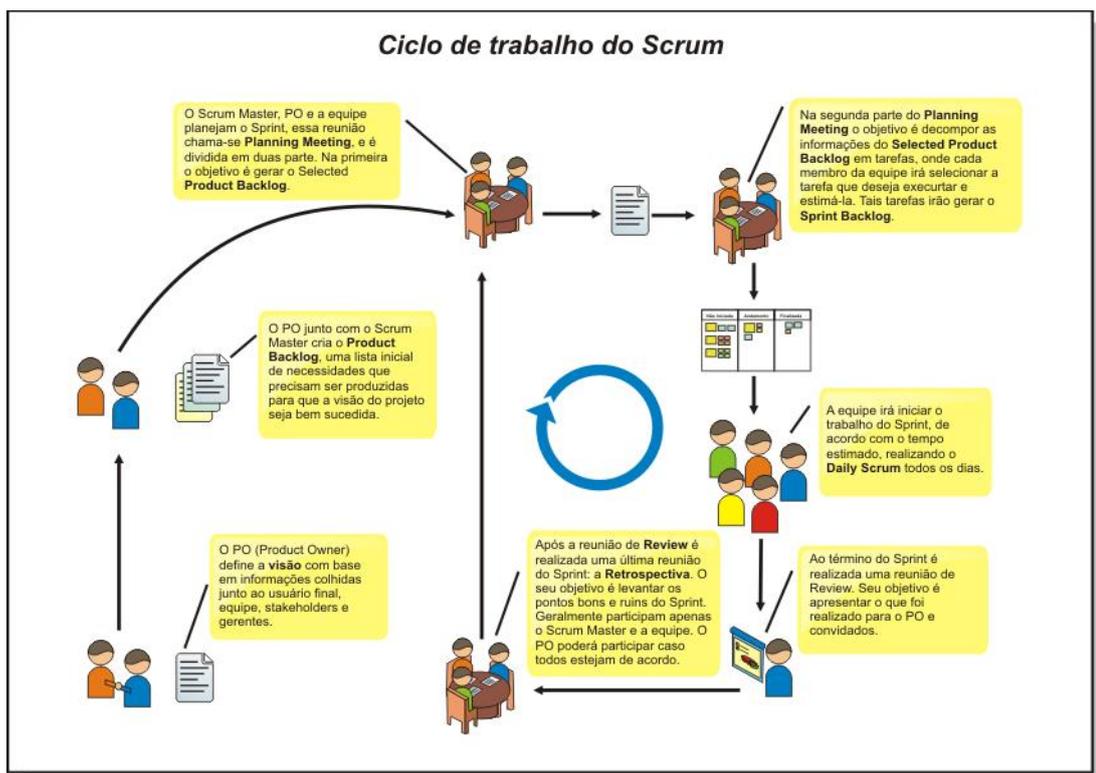


Figura 5 – Ciclo de trabalho do Scrum
Fonte: Moraes (2010)

3.2.1 Papéis

Os papéis, ou *roles* são representados por três entidades principais, o Proprietário do Produto, o *ScrumMaster* e o Time de trabalho.

O trabalho inicia-se com a escolha e definição do produto, seus requisitos, qual o real interesse do cliente que será representado pela entidade chamada de *ProductOwner*, o proprietário do Produto que segundo Schwaber (2011) é o responsável por maximizar o valor do produto e trabalho da equipe de desenvolvimento. Suas atribuições são definir conteúdo, características, data de término, rentabilidade, funções prioritárias de acordo com o que for de interesse do cliente e mercado, efetuar o reajuste de tarefas e recursos a cada trinta dias caso seja necessário e aceitar ou rejeitar os seus resultados.

A entidade responsável por liderar a equipe de desenvolvimento é chamada de *ScrumMaster*, que é responsável por garantir que o Scrum seja entendido e aplicado (SCHWABER, 2011) . Este deve ser a ponte entre o proprietário do produto e os desenvolvedores, tem como tarefas garantir a produtividade da equipe incentivando sua cooperação, protegendo-os de influências externas, conflitos internos e seguir o cronograma proposto pela metodologia. Pode ser chamado a grosso modo de Coordenador do projeto, além das tarefas envolvendo a equipe, é responsável por desempenhar três principais responsabilidades.

Acompanhar o desenvolvimento e conclusão de cada atividade, analisando possíveis mudanças que possam ocorrer no curso do desenvolvimento, bem como garantir que tarefas não fiquem paradas gerando ociosidade para a equipe.

Deve avaliar e conter possíveis bloqueios que causem prejuízo ao projeto, definindo prioridades e organizando as equipes como times para propor soluções e sanar problemas.

Na questão humana do desenvolvimento o *ScrumMaster* é responsável por resolver conflitos entre integrantes do time de desenvolvimento, mantendo um diálogo amplo para unificar a equipe e elucidá-los sobre a importância do trabalho em equipe.

A equipe de desenvolvimento são os programadores em si, variando da complexidade do sistema a ser desenvolvido podem existir várias equipes. Cada equipe tem em média sete participantes que definem suas metas junto ao *ScrumMaster* e tem a função de atingir todos os resultados propostos pelo Proprietário do Produto.

3.2.2 Cerimônias

As reuniões que ocorrem dentro do período de desenvolvimento são chamadas de *cerimonies*, ou cerimônias, são subdivididas em três tipos, a reunião de planejamento do *Sprint* que é como é chamada cada etapa do ciclo de desenvolvimento. Nessa reunião o proprietário do produto se reúne com o cliente para definir todos os recursos que serão agregados ao projeto, bem como a prioridade de desenvolvimento de cada módulo.

Durante o desenvolvimento são realizadas também reuniões diárias entre o *ScrumMaster* e a equipe de desenvolvimento para verificar o andamento do projeto, nessa reunião, cada funcionário deve responder o que fez no dia anterior, o que fez no dia atual e quais as dificuldades surgiram durante o desenvolvimento. Nas reuniões diárias o *ScrumMaster* terá o papel de identificar possíveis problemas ou novas atividades que deverão ser adicionadas ao desenvolvimento e planejar a sua resolução ou construção.

A cada final de período de *Sprint* é realizada uma reunião de revisão de *Sprint*, onde o *ScrumMaster* e a Equipe de trabalho se juntam ao Proprietário do Produto, com ou sem o cliente para apresentar o resultado do *Sprint*, ou seja, o que foi desenvolvido durante aquele ciclo. Ao final da apresentação, são definidas quais as prioridades do cliente para criar o próximo *Sprint*. Numa segunda parte da reunião, entre *ScrumMaster* e equipe de desenvolvimento, são analisados os pontos positivos e negativos encontrados na apresentação e no desenvolvimento do *Sprint*.

3.2.3 Artefatos

Os artefatos do Scrum representam o trabalho ou o valor das várias maneiras que são úteis no fornecimento de transparência e oportunidades para inspeção e adaptação (SCHWABER, 2011).

São as ferramentas de trabalho do modelo de desenvolvimento as ferramentas são: O *ProductBacklog* que é a lista de funcionalidades e estimativa de custo de desenvolvimento, é desenvolvido no começo do projeto na reunião entre o proprietário do produto e o cliente. No *ProductBacklog* são listadas funcionalidades visíveis a cliente e também os requisitos técnicos para desenvolvimento e execução do projeto.

Outro artefato é o chamado *Sprint Backlog*, é a subdivisão do *ProductBacklog* organizada de acordo com a prioridade de desenvolvimento, como uma lista do que é necessário para desenvolver cada parte do projeto.

O *Burndown Chart* é o terceiro artefato do *Scrum*, é um gráfico que demonstra o trabalho de cada *Sprint*, bem como o que foi acumulado durante cada *Sprint*, a altura do gráfico indica a quantidade de tarefas envolvidas no *Sprint Backlog* que não foram completadas e o comprimento no gráfico indica os dias. Facilitando assim a visualização do tempo em função das tarefas, na metodologia *Scrum* o tempo médio de cada *Sprint* é de cerca de trinta dias.

3.3 CONSIDERAÇÕES SOBRE O CAPÍTULO

Através deste capítulo, foram constatadas as premissas do método de desenvolvimento ágil *Scrum*, além do levantamento da teoria que possibilitou a criação desse método, seus desenvolvedores e passos necessários para executar o método em si.

Através desse levantamento foi avaliada como viável a utilização do método para o desenvolvimento do sistema ao qual este trabalho se refere pois as conclusões sobre desenvolvimento ágil aqui representado pelo *Scrum* foram positivas.

4 PROTÓTIPO

Mensurar uma atividade e atribuir valor a ela é um conceito complexo, pois além de contar com os mais diversos fatores referentes a atividade desempenhada ainda conta com fatores internos a empresa onde o funcionário presta seu serviço, externos como empresas do mesmo ramo com funções parecidas e até pessoais onde a relação de amizade de determinado funcionário com o seu superior influencia diretamente em sua remuneração.

Os planos de cargos e salários são métodos utilizados para contribuir nessa tarefa de determinar o valor de cada função de acordo com fatores estatísticos, pessoais e profissionais. Este trabalho propõe uma ferramenta para auxiliar na coleta dos dados envolvidos no processo de implantação de um plano de cargos e salários e desse modo tornar mais eficiente o levantamento e a análise das informações. Para guiar o desenvolvimento foi selecionada a metodologia Scrum e como ferramenta de auxílio ao desenvolvimento o *framework* Jaguar.

O capítulo está dividido em quatro seções, onde a seção 4.1 faz uma descrição do ambiente e da aplicação que foi desenvolvida. A seção 4.2 descreve a utilização do *framework* Jaguar no desenvolvimento do sistema, a seção 4.3 apresenta como a metodologia Scrum foi aplicada para a implementação do *software* e a sessão 4.4 onde são apresentadas as considerações sobre o capítulo.

4.1 DESCRIÇÃO

O objetivo principal da aplicação é aumentar a eficiência dos processos de coleta e relacionamento dos dados envolvidos na elaboração de planos de cargos e salários.

Por meio de um ambiente web a ferramenta fornece formulários para entrada de dados sobre a empresa, setores, funcionários, cargos, atribuições de cada cargo e de acordo com cada método, um nível de importância para cada um dos itens cadastrados.

Por ser uma ferramenta de auxílio ao cadastro das informações, ela não obedece a um método específico de elaboração de planos de cargos e salários, podem ser criados módulos para entrada de informações de acordo com o método

que cada empresa estiver disposta a utilizar. Para o fim de melhor demonstrar a ferramenta seu desenvolvimento neste trabalho foi voltado para o método dos pontos, por contar com dados mais mensuráveis graças ao seu sistema de pontuação.

A ferramenta pode funcionar num servidor interno a empresa no conceito de intranet ou *online* pela internet. O funcionamento básico é lustrado na Figura 6.

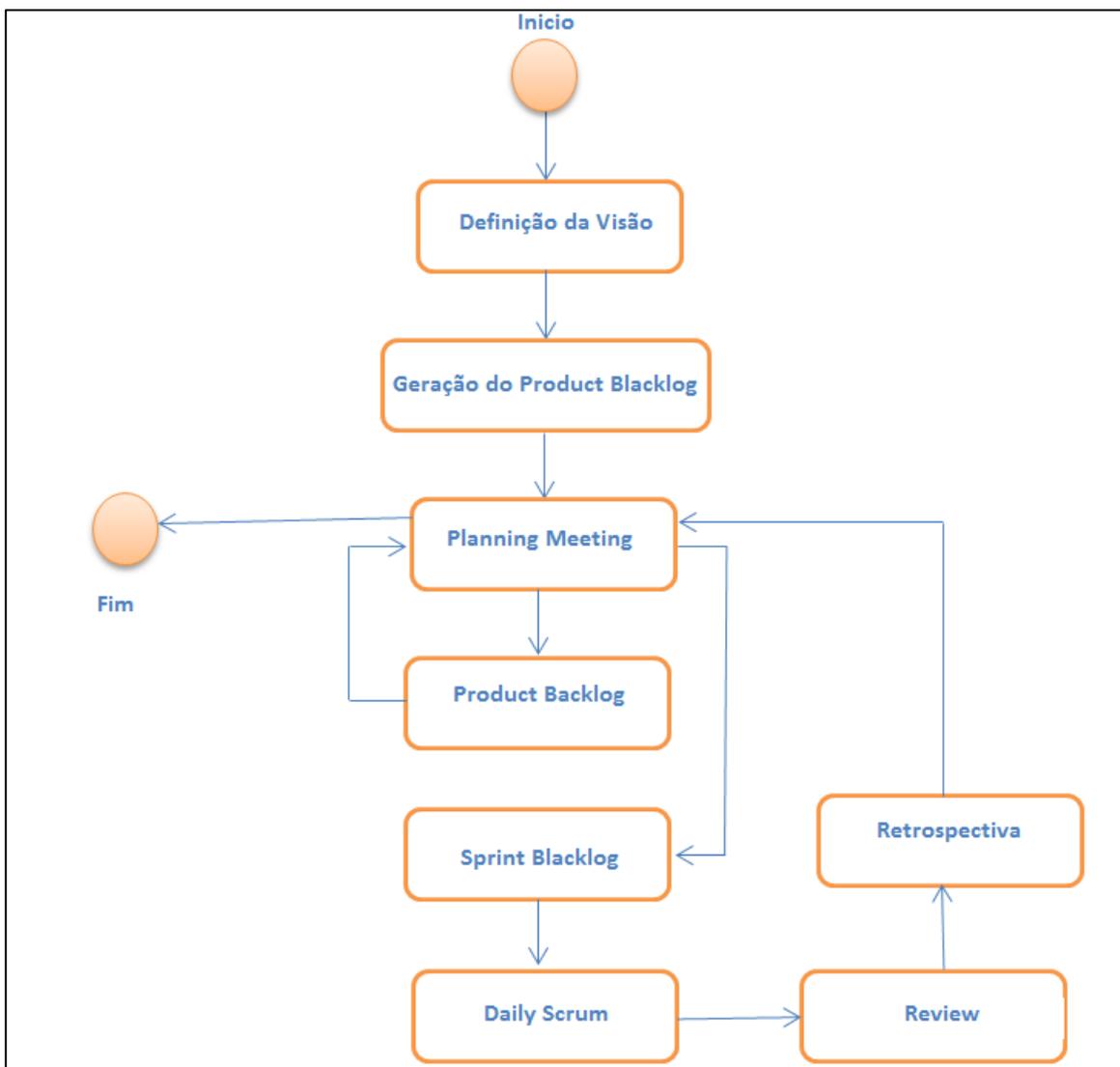


Figura 6 – Diagrama de blocos
Fonte: Autoria própria.

Fornecer diferentes cadastros para entrada de dados, num primeiro momento a interface de Cadastro de Empresa relaciona itens de identificação da corporação onde o *software* será aplicado como descrição e dados de identificação. Uma interface de cadastro de cargos permite incluir as atividades existentes na empresa,

sua remuneração, hierarquia e as atribuições de cada uma. Trabalhando em conjunto com a metodologia dos pontos, seus requisitos mentais, físicos, responsabilidades e descrição do ambiente em que o trabalho é desempenhado para aquela função definindo os pontos para cada diretriz. No cadastro de cargos também deve ser levada em consideração a consistência externa, ou seja, também é listada a média salarial de outras empresas do ramo considerando para tal a pesquisa de três empresas com ramo de atuação semelhante e cargos semelhantes.

A interface de cadastro de funcionários fornece a entrada de dados pessoais e profissionais e relaciona cada funcionário com seu cargo já descrito no cadastro de cargos.

Depois de realizada a entrada de dados a ferramenta disponibiliza diversos relatórios que podem relacionar os funcionários com seus cargos, hierarquia, atribuições e classificá-los por salário, setor, tempo de serviço e demais dados descritivos.

O cenário idealizado para implementação da ferramenta é de uma empresa de médio porte que possua setor de recursos humanos e um funcionário dedicado a implantar o plano de cargos e salários, a tarefa de analisar as informações relacionadas no *software* ficará a cargo desse funcionário e seus superiores, pois inicialmente não é prevista nenhuma introdução de inteligência artificial para que o *software* possa tomar decisões por conta própria, ou seja, o *software* apenas auxilia o processo de coleta, demonstração e relacionamento das informações, as decisões tomadas a partir dos dados coletados são de responsabilidade da empresa que o adquirir.

4.2 O *FRAMEWORK* JAGUAR NO DESENVOLVIMENTO

Dentre os fatores determinantes para a escolha do *framework* Jaguar para desenvolvimento da ferramenta está o fato de ser um *framework* que trabalha com plataformas de código aberto, que além de não gerar ônus para aquisição do *software*, ainda contribui para que exista uma comunidade focada em compartilhamento de conhecimentos e auxílio a desenvolvedores que estão iniciando com o desenvolvimento no *framework* Jaguar.

Outro fator culminante para a escolha do Jaguar foi o fato do *framework* permitir uma liberdade maior em relação ao banco de dados, pois a criação do banco é automatizada de acordo com a inserção dos formulários e definição de seus tipos e atributos.

4.3 A METODOLOGIA SCRUM NO DESENVOLVIMENTO

A metodologia Scrum foi selecionada como ferramenta de auxílio ao desenvolvimento do projeto dada a sua capacidade gerencial diante de outras metodologias focadas mais exclusivamente na programação em si.

Essa capacidade gerencial empregada no trabalho proposto fez com que o Scrum fosse de grande ajuda desde a fase de levantamento de informações e requisitos básicos até a ordenação dos processos, cronogramas e conclusão do objetivo proposto.

Como ferramenta para nortear o desenvolvimento padronizado de acordo com a metodologia Scrum foi utilizado o Scrumhalf, um *website* que gerencia equipes e projetos baseados em Scrum.

O uso da ferramenta Scrumhalf é iniciado com o cadastro dos participantes no *website* que disponibiliza planos de assinatura para equipes e projetos maiores e também uma opção grátis utilizada neste trabalho que permite o gerenciamento de um projeto com até três membros.

Após o cadastro, o usuário visualiza um painel de administração onde pode criar seu projeto e convidar outros usuários para tornarem-se membros ou aguardar o convite de outros usuários para seus respectivos projetos.

O usuário que cria um projeto torna-se administrador daquele projeto, tendo a permissão para editar as opções no painel de administração referentes ele, convidar outros membros e definir suas atribuições. A Figura 7 apresenta o painel de administração do projeto referente a este trabalho.

scrumhalf TCCJaguar :: Cadastros do Sistemas

PROJETO PRODUCT BACKLOG SPRINT USUÁRIO PLANOS SUPORTE

SUMÁRIO RELATÓRIOS GPE

DESCRIÇÃO

Nome: TCCJaguar
 Data Início: 01/03/2012
 Data Início Game: 15/05/2012
 Nome Administrador: Vilmar Rogoski Junior
 Descrição: Desenvolvimento do software plano de cargos com o framework jaguar.
 Definição de história preparada: Caso uso definidos.
 Definição de história pronta: Documentação de requisitos pronta.

PARTICIPANTES

NOME	SIGLA	PAPÉIS	
Vilmar Rogoski Junior	VRJ	SM	[Editar] [Excluir]
Evaldo Andrade dos Anjos	EADA	PO	[Editar] [Excluir]
Leandro Alves de Fontes Silva	LADFS	ES	[Editar] [Excluir]

CONFIGURAÇÕES

WebSite: www.scrumhalf.com.br
 Twitter:
 Velocidade Inicial: 15
 Timebox Sprint (semanas): 1
 Idioma: Português

[Editar] [Excluir]

Figura 7 – Painel de administração do Scrumhalf
 Fonte: Autoria Própria

O painel de controle da Figura 7 apresenta na tabela da esquerda a descrição do projeto, onde pode-se definir nome, data de início, fim, descrição e especificidades do projeto. Na tabela da direita, encontram-se as opções referentes aos participantes convidados, e também é possível convidar outros usuários na opção novo, remover usuários do projeto e através do botão editar mudar suas atribuições, onde a sigla SM representa o *Scrum Master*, a Sigla PO o *Product Owner* e a sigla ES a Equipe de *Software*.

A tela de *Product Backlog* apresentada na Figura 8 apresenta as histórias como definido pelo *Scrumhalf*, que são propostas apresentadas pelos membros do projeto com um título e uma breve descrição para serem incluídas durante o desenvolvimento para serem aprovadas, ou não pelo *Scrum Master* e pelo *Product Owner*, no *Product Backlog* da *ScrumHalf* as histórias se dividem em propostas, aprovadas, e recusadas.

ID	TÍTULO	VALOR AGREGADO	ESTIMATIVA
15	Implementação do caso de uso "Manter Município".	50	2,0
16	Implementação do caso de uso "Manter Setor".	50	2,0
18	Implementação do caso de uso "Manter Unidade Organizacional".	50	2,0
19	Implementação do caso de uso "Manter Endereço".	50	2,0
20	Implementação do caso de uso "Manter Histórico Profissional".	80	2,0
22	Implementação de validações de campos.	60	2,0
23	Implementação de recursos RIA.	50	2,0
24	Customização layout.	50	2,0

Figura 8 – Interface do Product Backlog
Fonte: Autoria Própria

No desenvolvimento da ferramenta de apoio ao plano de cargos e salários foram propostas e aprovadas as tarefas de:

- Implementação do Caso de Uso "Manter UF";
- Implementação do caso de uso "Manter Município";
- Implementação do caso de uso "Manter Setor";
- Implementação do caso de uso "Manter Cargo";
- Implementação do caso de uso "Manter Unidade Organizacional";
- Implementação do caso de uso "Manter Endereço";
- Implementação do caso de uso "Manter Histórico Profissional";
- Implementação do caso de uso "Manter Funcionário";
- Implementação de validações de campos;
- Implementação de recursos RIA;
- Customização do *layout*.

As tarefas de caso de uso relacionadas como manter UF, Município, Setor, Cargo, Unidade Organizacional, Endereço, Histórico profissional e Funcionário,

foram incluídas posteriormente como histórias do *Scrum* no *Product Backlog* para realizar as operações do CRUD na ferramenta. O Diagrama de caso de uso do módulo manter setor é apresentado na Figura 9.

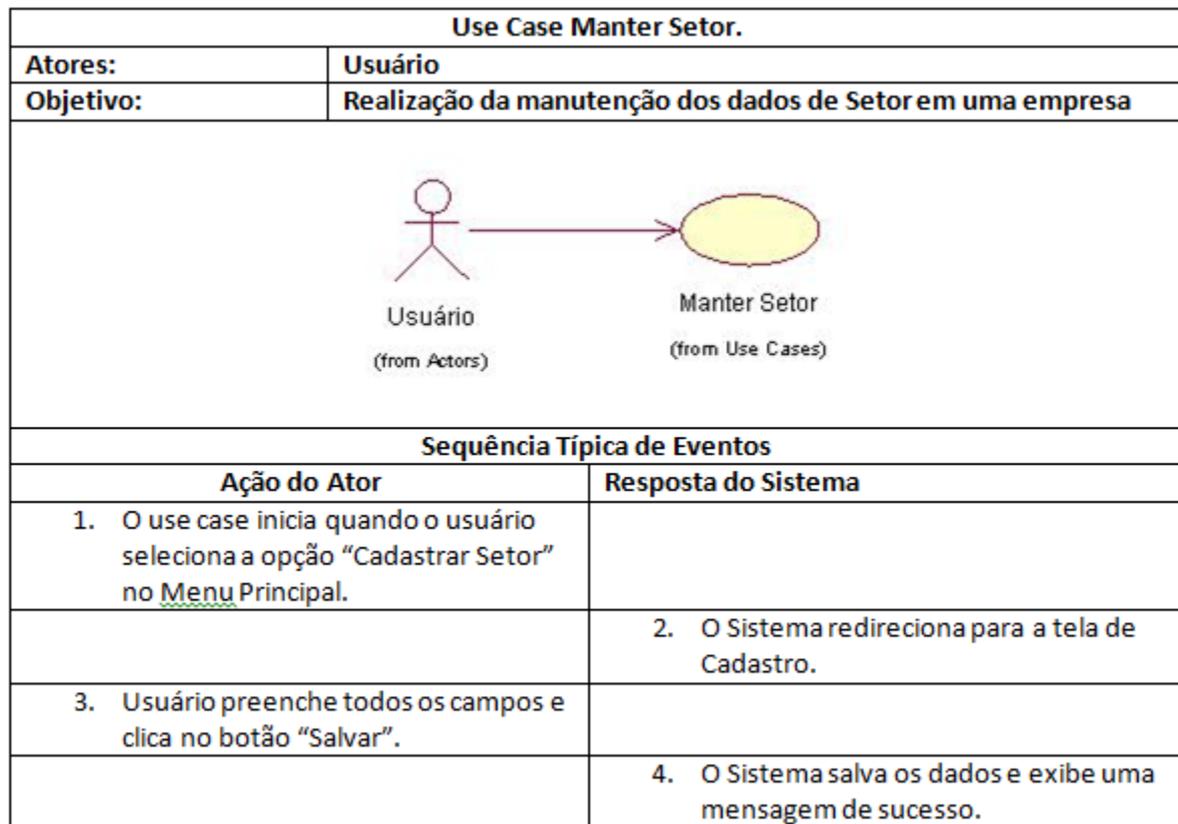


Figura 9 – Use case Manter Setor
Fonte: Autoria Própria

Outra tarefa assim adicionada foi a de validação dos campos para levantamento de informações como localização através do CEP. Os recursos de RIA para inclusão de plataformas auxiliares no desenvolvimento das páginas e por fim a customização do layout onde foi criada uma interface de usuário diferente dos padrões já fornecidos junto ao *framework*.

A ferramenta ScrumHalf disponibiliza no menu *Sprint* a definição da *Sprint* atual, onde é apresentada a meta referente à aquela *Sprint*, bem como sua previsão de conclusão e seu status, que pode ser prevista, quando está no planejamento, porém não foi iniciada, em andamento quando a equipe já está trabalhando na referida tarefa, em revisão quando o desenvolvimento foi concluído, porém ainda não foi revisado, em retrospectiva quando em aguardo da reunião de retrospectiva do *Sprint* e finalizada quando todas as rotinas envolvidas já foram realizadas.

O menu *Sprint* também apresenta um quadro de tarefas, onde é exibido o andamento de cada uma das histórias aprovadas, o quadro exibe informações como a data e situação de determinada tarefa em cada *Sprint*.

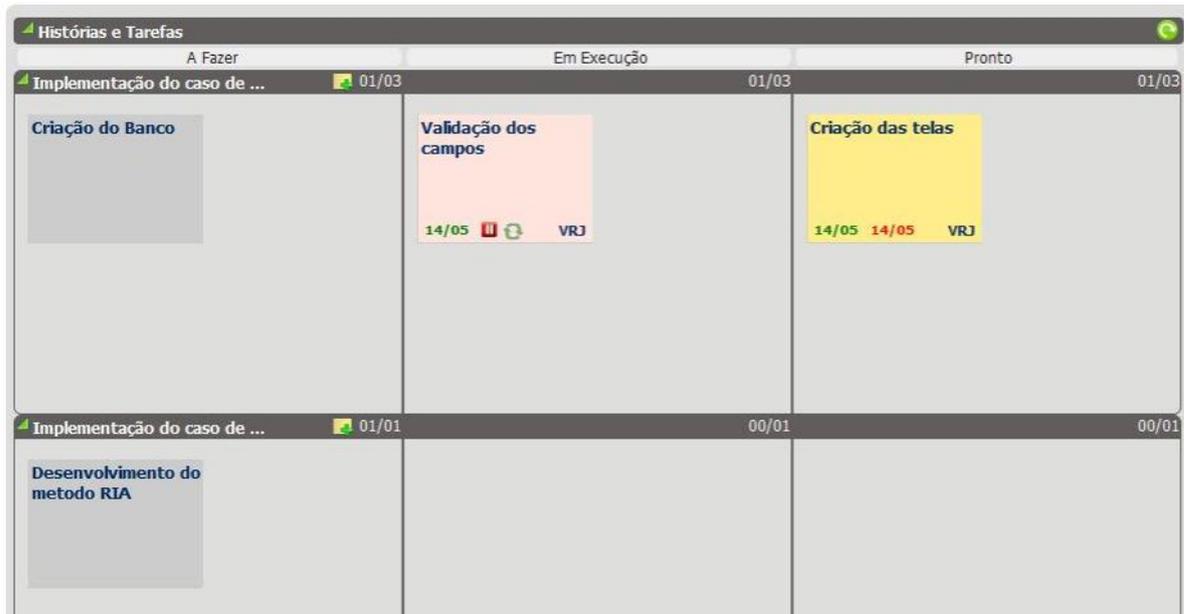


Figura 10 – Quadro de tarefas
Fonte: Autoria Própria

A Figura 10 apresenta uma tarefa associada à implementação do caso de uso “manter cargo”, as cores de cada quadro no calendário de desenvolvimento tem funções distintas, onde o amarelo representa uma tarefa considerada normal, o vermelho representa uma falha e o cinza um impedimento, seja por uma falha ou por uma interdependência com outra tarefa ainda não realizada.

4.4 CONSIDERAÇÕES SOBRE O CAPÍTULO

Conclui-se neste capítulo que a forma de documentar, extrair, administrar e criar durante o desenvolvimento de um projeto é através da escolha das ferramentas adequadas.

Por meio da utilização da metodologia ágil Scrum e *framework* Jaguar houve uma contribuição na confecção e decisões referentes ao protótipo, minimizando situações errôneas.

5 DESENVOLVIMENTO DA APLICAÇÃO

Este capítulo apresenta os passos envolvidos no desenvolvimento do módulo de cadastro de setor utilizando o *framework* Jaguar, posteriormente são apresentados os resultados obtidos no desenvolvimento bem como as considerações finais.

5.1 ETAPAS DO DESENVOLVIMENTO

Tendo como objetivo principal a demonstração do funcionamento do *framework* Jaguar no desenvolvimento da aplicação, nos parágrafos a seguir é demonstrada a criação do módulo de cadastro de setor onde o usuário do *software* efetua a inclusão dos setores existentes na empresa que utiliza a ferramenta para implantação do plano de cargos e salários.

O desenvolvimento iniciou-se com o levantamento dos requisitos básicos para desenvolvimento dos módulos que seriam independentes ao modelo de plano de cargos e salários selecionados, ou seja, atributos que seriam comuns a todos os modelos como dados da empresa, setores, cadastro de funcionários e cadastro de cargos.

O *framework* Jaguar funciona através de uma integração com a IDE do Eclipse, onde as funcionalidades do Jaguar passam a fazer parte da interface de desenvolvimento do Eclipse.

Através da opção *New Project* do menu do Jaguar é possível acessar as opções para criação de um projeto. Para a criação dos módulos como o cadastro de setor, foi utilizada a opção “Criar novo projeto Jaguar” como na Figura 11.

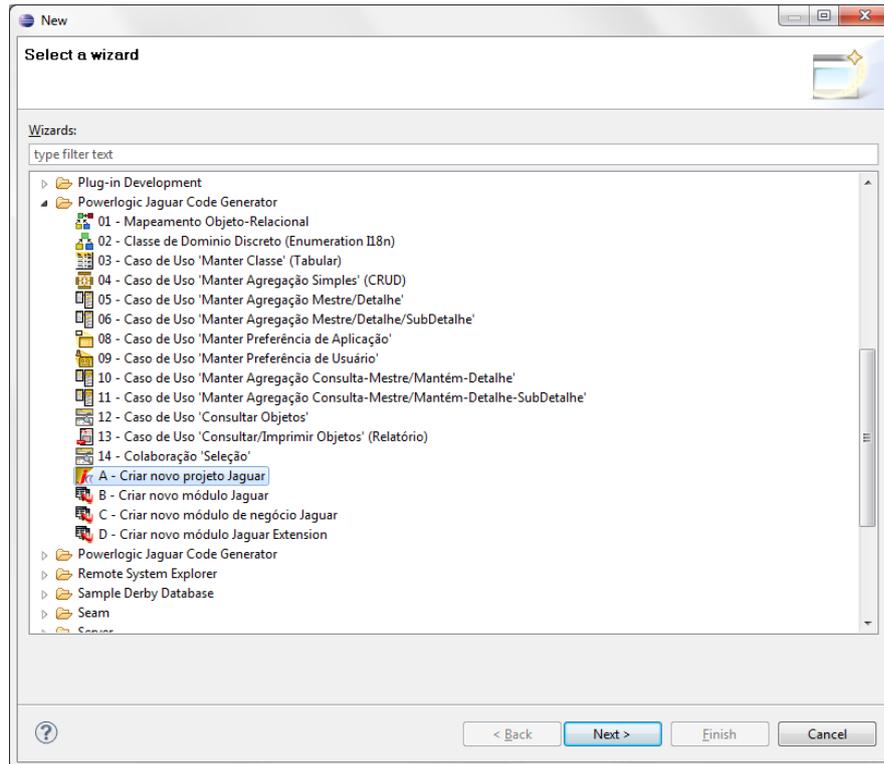


Figura 11 – Tela para novo projeto no Jaguar
Fonte: Autoria própria

A primeira tela do assistente de criação apresentada na Figura 12 solicita o Nome do projeto, seu diretório e a pasta com os *templates*, ou seja, o local onde serão armazenadas as informações referentes ao MVC que divide a parte lógica da aplicação da interface do usuário. Por fim, o assistente propõe que o usuário selecione a opção Projeto Simples que separa o MVC em pacotes, facilitando o trabalho com a interface em um projeto único ou Projeto Distribuído, onde o MVC é fracionado em projetos, ideal para utilização do mesmo padrão de interface por projetos diferentes. No desenvolvimento da ferramenta em questão foi selecionada a opção de Projeto Simples devido à existência de apenas um projeto a ser desenvolvido.

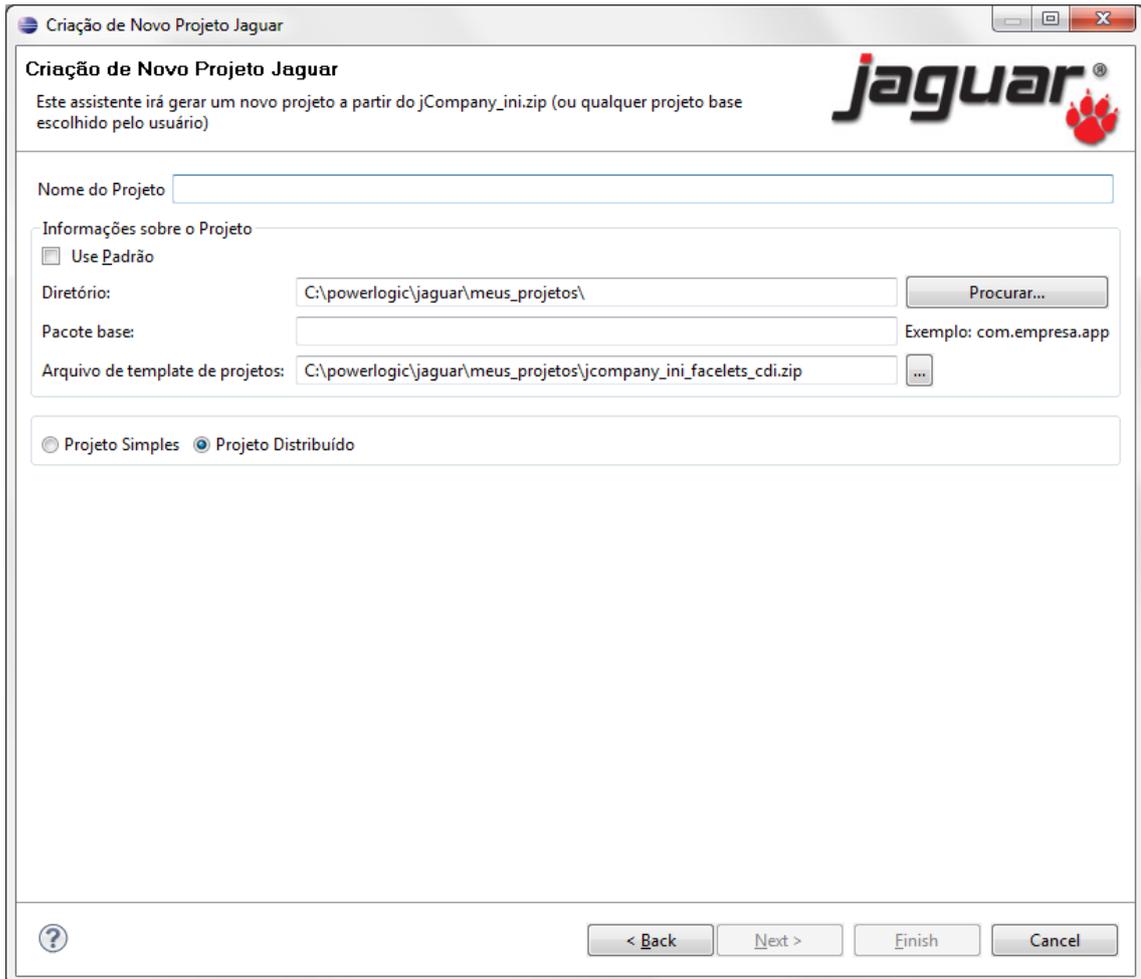


Figura 12 – Assistente de criação de projeto do Jaguar
Fonte: Autoria própria

O gerenciador de banco de dados já é instalado automaticamente junto ao Jaguar através da ferramenta *Apache Derby*, a associação da ferramenta ao projeto é feita através do menu de contexto do projeto através da opção *ADD Apache Derby nature*. Após a associação, o menu de contexto do projeto apresenta a opção de inicialização do banco de dados, como demonstrado na Figura 13.

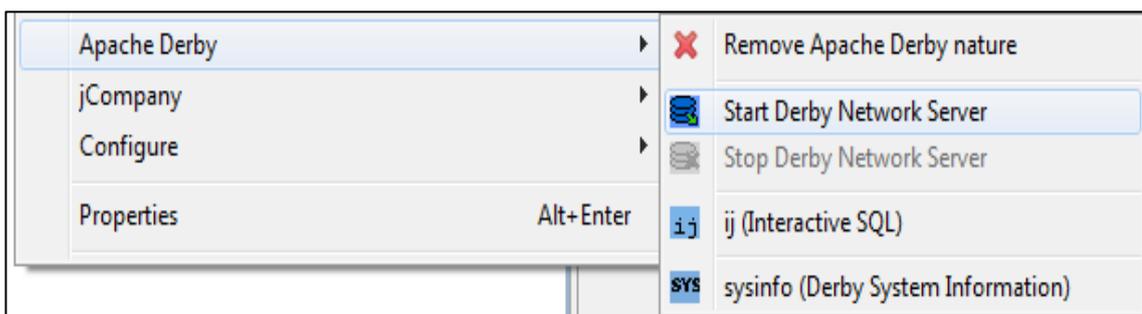


Figura 13 – Inicialização do Apache Derby
Fonte: Autoria própria

A criação de uma nova classe num projeto do Jaguar apresentada na Figura 14 é realizada no diretório referente ao nome do projeto, neste caso `jagtcc_commons` na pasta Java dentro de `main`, por sua vez dentro da pasta `src`, para tal é utilizado o pacote `entity`. A classe criada na Figura 14 foi denominada `setor`.

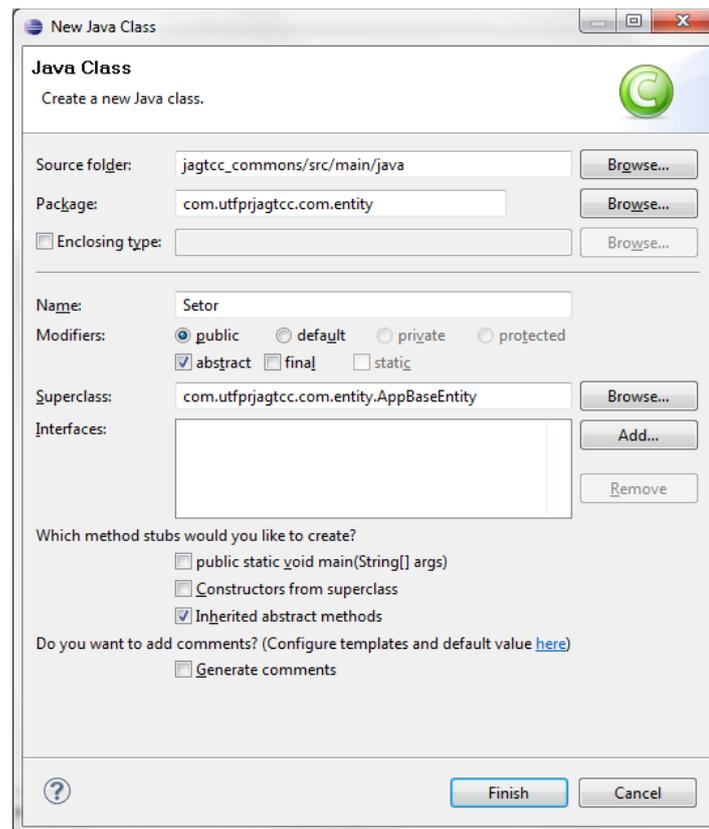


Figura 14 – Criação da classe Setor
Fonte: Autoria própria

Junto à criação da classe `Setor` que na Figura 15 contém os campos `ramal` e `descricao` é herdada a classe `AppBaseEntity` onde três propriedades já são pré-mapeadas.

A propriedade `versão`, representada por `@version` implementa um sistema de controle de versões JPA baseada no princípio de tratamento de concorrência otimista, segundo (POWERLOGIC, 2011). Deste modo, se dois usuários estão alterando o mesmo registro simultaneamente, há garantia de que um não irá sobrepor atualização de outro, caso o segundo não esteja com a versão atualizada dos dados recuperados.

Já as propriedades `dataUltAlteracao` e `usuarioUltAlteracao`: implementam o que é chamado no Jaguar de “auditoria pauta mínima”. Seus valores são atualizados

pelo Jaguar para guardar o *login* do usuário que realizou a última alteração no objeto e também a data/hora desta modificação para fins de auditoria e segurança.

O Mapeamento das classes é realizado na opção de Mapeamento Objeto-Relacional, para criação do cadastro de setores, as opções atribuídas são demonstradas na Figura 15.

Propriedade	Estereótipo	Coluna	Obrig.	Lookup	Classe	Tabela FK	Tipo SGBD	Tamanho
ramal	property	RAMAL	Sim	<input checked="" type="checkbox"/>				
descricao	property	DESCRICA0	Sim	<input type="checkbox"/>				

Figura 15 – Mapeamento Objeto-Relacional
Fonte: Autoria própria

A opção “Gerar Entidades descendentes de não existirem” mostrada na Figura 15 foi marcada pois faz com que o gerador produza uma nova classe com nome padrão <Entidade>*Entity* ou utilizando o sufixo padrão para entidades definível no *framework* via anotações, como por exemplo: Para a classe *Setor.java*, será produzida uma classe adicional *SetorEntity.java*, concreta e descendente de *Setor.java*. A marcação “Informações Adicionais” faz com que uma segunda janela do Assistente seja apresentada com informações adicionais que possibilitam a geração de outras anotações do jaguar.

A lista de “Estereótipos” traz classificações de relacionamentos e natureza de Entidades, segundo padrões de Arquitetura do *framework*, que auxiliam na definição dos mapeamentos. O campo “Classe” traz a classe para mapeamento de acordo com o que foi determinado na criação do projeto.

No campo “Tabela” o Jaguar gera uma proposta para o nome da Tabela relacional do SGBD, inserindo nomes em maiúsculas e utilizando sublinhados como

separadores. Os nomes são apenas sugestões e podem ser alterados pelo desenvolvedor.

A lista “Identificador” traz diferentes estratégias para identificação de objetos, no projeto, foi utilizada a opção `OID-Auto` onde o *framework* gera um mapeamento para uso padrão cuja identificação é realizada via a propriedade `privateLong id`, com valores gerados automaticamente pelo gerenciador de banco de dados.

O campo “*Sequence*” permite que se informe o nome para criação de uma estrutura sequencial utilizada em alguns SGBDs relacionais como o Oracle, para geração de valores de identificação. O Padrão do jaguar é utilizar um “*Sequence*” por “Tabela” e com nome “`SE_<nome tabela>`”. Este nome também pode ser alterado pelo desenvolvedor. No caso da utilização de um gerenciador de bancos de dados não sequenciais, o campo é descartado.

A coluna “Propriedade” traz as propriedades das classes candidatas a serem persistidas. A coluna “Estereótipo” define classificações de mapeamentos típicos utilizados no jaguar. A coluna “Coluna” apresenta nomes sugeridos para as propriedades das classes que, tal como no caso da Tabela, podem ser modificados .

A coluna “Obrigatório” permite que se marque se a Coluna gerada na Tabela relacional será gerada com “`NOT NULL`” (Obrigatório = Sim) ou “`NULL`” (Obrigatório = Não).

A coluna “*Lookup*” permite que se selecione uma ou mais colunas para comporem a cláusula do método padrão “*toString*”

A coluna “Classe” é utilizada em relacionamentos *manyToOne*, *oneToMany* e variações,contendo o nome da classe referenciada por propriedades ou coleções.

A coluna “Tabela FK” exhibe um nome que será utilizado durante a geração de DDL automatizada do Jaguar, especificamente para nomear Chaves Estrangeiras (*Foreign Keys*) quando mapeando relacionamentos.

A coluna “Tipo SGBD” exhibe o tipo que será utilizado para a geração da DDL (*Data DefinitionLanguage*), código que define tabelas para um SGBD relacional. A Figura 16 apresenta um trecho do código referente as operações da classe setor criadas automaticamente pelo *framework*.

```

import com.powerlogic.jcompany.domain.validation.PlcUnifiedValidation;

@MappedSuperclass
@PlcUnifiedValidation
public abstract class Setor extends AppBaseEntity {

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO, generator = "SE_SETOR")
    @Column(nullable=false,length=5)
    private Long id;

    @NotNull
    @Size(max = 10)
    @Column
    @PlcReference(testDuplicity=true)
    private String ramal;

    @NotNull
    @Size(max = 40)
    @Column
    private String descricao;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id=id;
    }

    public String getRamal() {
        return ramal;
    }

    public void setRamal(String ramal) {
        this.ramal=ramal;
    }
}

```

Figura 16 – Código da classe Setor
Fonte: Autoria própria

O desenvolvimento da parte visual, ou seja a visão e as telas são realizadas também através de assistente, por meio do menu “new” e escolha da opção “03 – Caso de uso ‘ Manter Classe’ (tabular)” . A Figura 17 representa a tela com as opções para criar a visão.

Caso de Uso 'Manter Classe'

Geração de Caso de Uso 'Manter Classe' - Pr 

Este assistente irá gerar artefatos JSF2, Facelets e Mensagens em conformidade com o Caso de Uso 'Manter'

Projeto: jagtcc

Caminho Absoluto da Entidade Raiz/Principal: C:\powerlogic\jaguar\meus_projeto

Identificador do Caso de Uso (ex: notafiscal): setor

Nome do Pacote (ex: com.empresa.app.entity): com.utfprjagtcc.com.entity

Classe de Controle (ex: NotaFiscalMB): com.utfprjagtcc.com SetorMB

Classe de Fachada (ex: NotaFiscalFacadeImpl): com.utfprjagtcc.com PlcFacadeImp

Classe de Modelo (ex: NotaFiscalRepository): com.utfprjagtcc.com PlcBaseRepos

Classe de Persistência (ex: NotaFiscalDAO): com.utfprjagtcc.com SetorDAO

Subdiretório: setor

< Back Next > Finish Cancel

Figura 17 – Assistente de criação da visão
Fonte: Autoria própria

Identificador do Caso de Uso (URL) é o prefixo para o Jaguar gerar a URL, nome base de JSPs, XHTML e demais extensões seguindo convenções de nomenclatura da Arquitetura.

O campo Entidade permite que se selecione a Entidade raiz da Agregação envolvida, cujas propriedades serão utilizadas como base para a proposta de um formulário e cujas instâncias serão persistidas, segundo a arquitetura MVC.

A Classe de Controle é utilizada caso seja prevista a necessidade de implementação de programações nesta camada para o Caso de Uso. Subdiretório é o diretório onde ficará a página, ou seja, a pasta que armazena seus arquivos.

Ao clicar na opção *Next* é apresentada a página do assistente referente ao título da página e suas opções de navegabilidade. Figura 18

The screenshot shows a window titled 'Caso de Uso 'Manter Classe'' with the Jaguar logo in the top right corner. The main heading is 'Geração de Caso de Uso 'Manter Classe' - Projeto: jagtcc'. Below this, a subtitle reads: 'Este assistente irá gerar artefatos JSF2, Facelets e Mensagens em conformidade com o Caso de Uso 'Manter Classe' do jCompany'. The window is divided into two main sections for configuration options.

Grupo Manutenção:

- Título (ex: Manter Nota Fiscal):
- Opções para geração de Manutenção:
 - Usar tecnologia Ajax para a Manutenção
 - Entrada em Lote
 - Tipo Exclusão: Lógica Física

Opções globais para geração:

- Gerar Metadados
- Gerar Mensagens(Resources)
- Gerar Itens de Menu
- Gerar arquivos de Página

At the bottom of the window, there are four buttons: a help icon (?), '< Back', 'Next >', 'Finish', and 'Cancel'.

Figura 18 – Opções da página
Fonte: Autoria própria

A próxima tela do assistente exibe as opções de manutenção da página. A Coluna “Usa?” define quais campos serão utilizados para realizar uma pesquisa na tela de cadastro, conforme ilustrado na Figura 19.

Caso de Uso 'Manter Classe'

Este assistente irá gerar o arquivo XHTML de argumento, seleção e manutenção conforme a especificação do usuário na tabela de opções

jaguar

Opções Globais:

Tipo Página XHTML: OO WYSIWYG Usar I18N

Opções de Visualização:

Usar Borda na Tabela Visualizar Pele: itunes Visualiza Manutenção

Propriedade	Usa?	Lin ...	Col M...	Formato	Domínio	Domínio de Classe	Rótulo	Identifi...
id	Sim	1	1	Inteiro	5		Cód.	
ramal	Sim	1	2	Texto	10		Ramal	
descricao	Sim	1	3	Texto	40		Descrição	
dataUltAlteracao	Não	1	1	Data	10		Data Ult. Alteração	
usuarioUltAlterac...	Não	1	1	Texto	10		Usuário Ult. Alter...	

Link Inteligente

Usar Link Inteligente Rótulo: Action:

< Back Next > Finish Cancel

Figura 19 – Assistente de manutenção de página
 Fonte: Autoria própria

O botão visualizar manutenção presente na Figura 19 permite uma prévia no navegador de internet. A Figura 20 apresenta esse recurso.

Cadastro de Setor

file:///C:/powerlogic/jaguar/meus_projetos/previewMan.html

Cadastro de Setor

	Exc.	Descrição	Ramal
1	<input type="checkbox"/>		
2	<input type="checkbox"/>		
3	<input type="checkbox"/>		

Figura 20 – Prévia do cadastro de setor
 Fonte: Autoria própria

A conclusão do assistente se dá pela opção *Finish* da Figura 19. Após concluir o assistente é necessário realizar a liberação para que o Tomcat disponibilize a página no servidor de internet. Após liberação o sistema poderá ser acessado através da URL <http://localhost:8080/np> onde “np” representa o nome do projeto.

Como o projeto representado foi nomeado “jagtcc” o acesso a ele é feito através do endereço: <http://localhost:8080/jagtcc/>.

O acesso ao sistema é realizado na tela de *login* da Figura 21, o usuário e a senha padrão do Jaguar são admin e senha, respectivamente. As credenciais de acesso são armazenadas num arquivo XML do próprio Tomcat chamado “tomcat-users.xml”.



Figura 21 – Tela de login
Fonte: Autoria própria

Com o usuário administrador devidamente conectado ao sistema a tela da Figura 18 é apresentada.

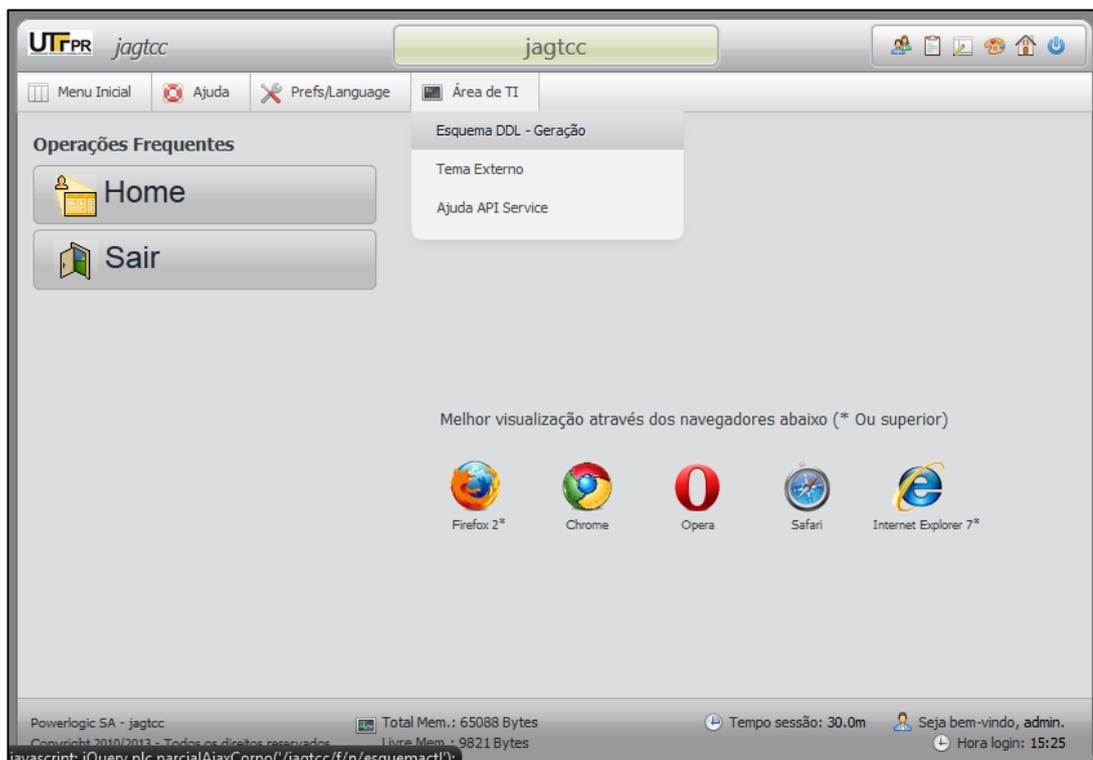


Figura 22 – Página inicial da administração
Fonte: Autoria própria

A opção “Área de TI” disponibiliza opções para manutenção da parte visual do projeto e Geração do Esquema DDL onde é gerada através de código SQL as instruções do banco de dados como na Figura 23.

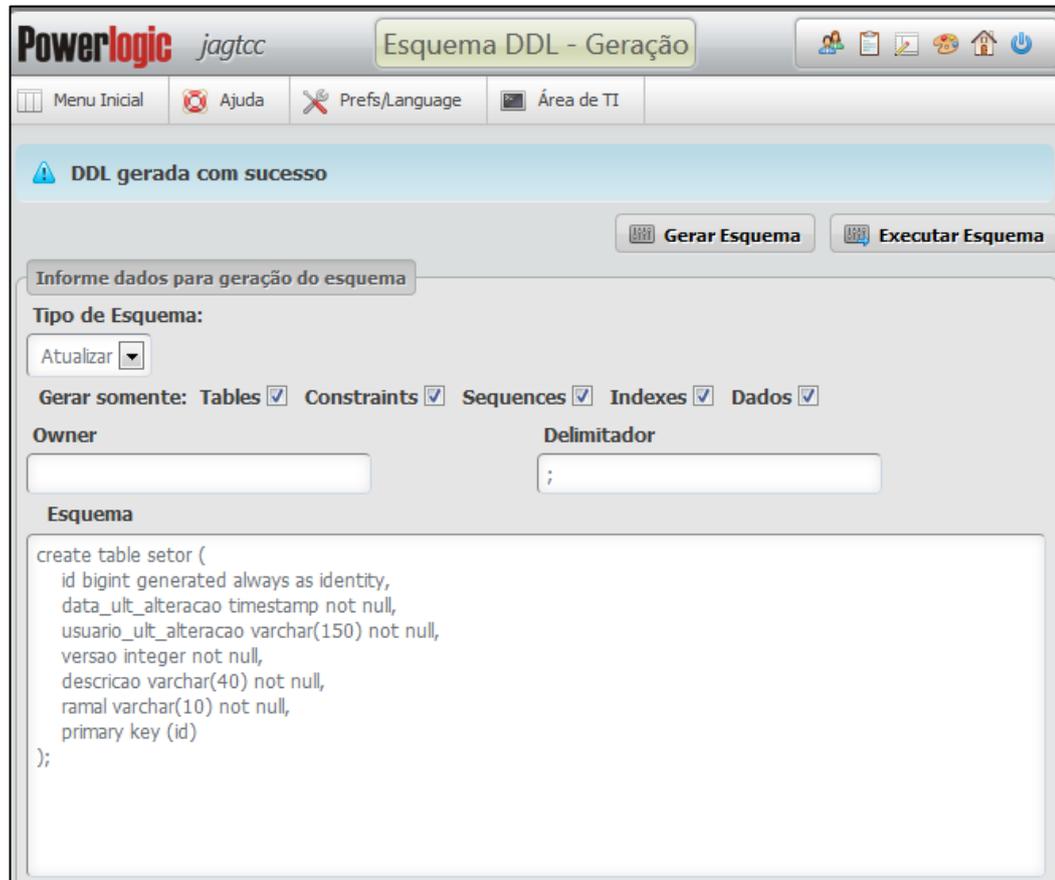


Figura 23 – Gerador de esquema de banco de dados
Fonte: Autoria própria

Após a geração do banco de dados o acesso à página já se torna disponível, nesse caso o cadastro de setor como na Figura 24.



Figura 24 – Tela do cadastro de setor.
Fonte: Autoria própria

5.2 RESULTADOS

Adotando como referência os objetivos determinados ao início do projeto os resultados se subdividem em quatro itens, a política de cargos e salários, o Jaguar como *framework* de desenvolvimento, o Scrum como metodologia ágil e o produto final da junção das duas ferramentas com a difícil tarefa de administrar salários.

A adoção de uma política de cargos e salários é uma tarefa complexa, pois se aplica justamente a um dos processos mais complexos no mercado de trabalho, a remuneração, fator determinado por uma linha tênue entre sucesso e fracasso empresarial, o plano de cargos e salários visa encontrar o equilíbrio entre motivar os funcionários a exercerem suas funções sem que a folha de pagamento seja um peso nas finanças da empresa.

Neste trabalho foi possível perceber a importância da tecnologia da informação numa tarefa de tamanha complexidade, pois com o sistema desenvolvido foi possível evitar o acúmulo de informações em meios físicos ou apenas em conversas informais, possibilitando a documentação digital de todos os procedimentos com mais segurança e agilidade, proporcionando também uma maior visibilidade do projeto como um todo através do relacionamento das informações coletadas de maneira automática e não analisando inúmeros documentos em busca de relações e fatores correspondentes.

A utilização do *Framework* Jaguar foi um norte de desenvolvimento e padronização que se revelou de grande importância para a criação do projeto, inúmeras telas, tabelas de bancos de dados, cadastros, menus, componentes e procedimentos envolvidos na criação da aplicação foram automatizadas no desenvolvimento com o *framework*, horas que seriam gastas com codificação de tarefas mais subjetivas como menus e interface ou mais importantes como cadastro, edição e modificação de registro puderam ser dedicadas a parte lógica da aplicação, aumentando assim o tempo útil gasto no projeto.

Já a metodologia Scrum se mostrou útil mesmo em se tratando de um projeto com um cliente fictício, visto que não se resume ao levantamento de requisitos com o cliente e um desenvolvimento mais genérico, e sim, de uma estrutura iterativa que além de levantar os requisitos, define de maneira objetiva quem realizará uma atividade, quando ela deverá estar pronta, quem supervisionará a tarefa, quais resultados devem ser obtidos e ainda propõe a discussão de todos estes itens entre os envolvidos no projeto. A interface do Sistema é bastante simples e amigável, em sua estrutura o Sistema é composto pelos módulos: Funcionários, UF, Município, Setor e Cargo.

The image shows the logo of the Universidade Tecnológica Federal do Paraná (UTFPR) at the top, consisting of the letters 'UTFPR' in a stylized font with a yellow square between the 'T' and 'F', and the full name 'UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ' below it. Below the logo is a login form with a light gray background. It contains three input fields: the first is labeled 'Login:', the second is labeled 'Senha:', and the third is a button labeled 'Ok'.

Figura 25 – Tela de Login
Fonte: Autoria própria

A tela de Login solicita o preenchimento de dois campos, para que o usuário informe seu Login e Senha para acesso ao Sistema.



Figura 26 – Tela Inicial
Fonte: Autoria própria

A Tela inicial é a porta de entrada do Sistema com o usuário. Através desta tela, o usuário terá acesso a todas as funcionalidades que o sistema oferece. Na tela inicial também recomendamos alguns navegadores para um melhor desempenho do sistema.

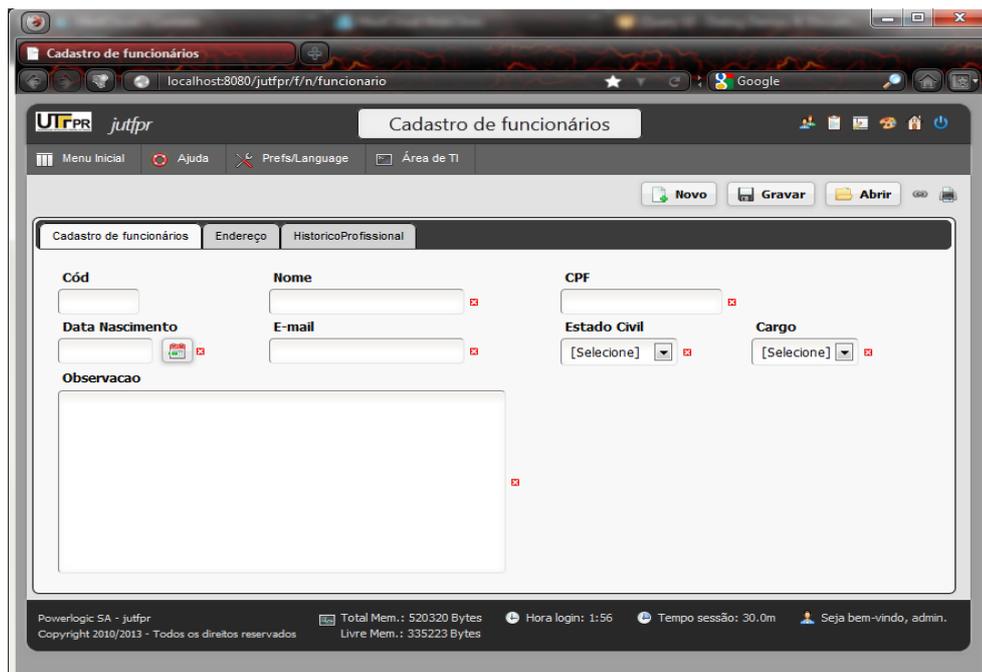


Figura 27 – Tela de Manutenção de Funcionários
Fonte: Autoria própria

O módulo de Funcionários permite gerenciar todos os dados dos funcionários que estão vinculados a um setor e cargo. As telas que compõe esse módulo são Cadastro de Pessoas, Endereço e Histórico Profissional.

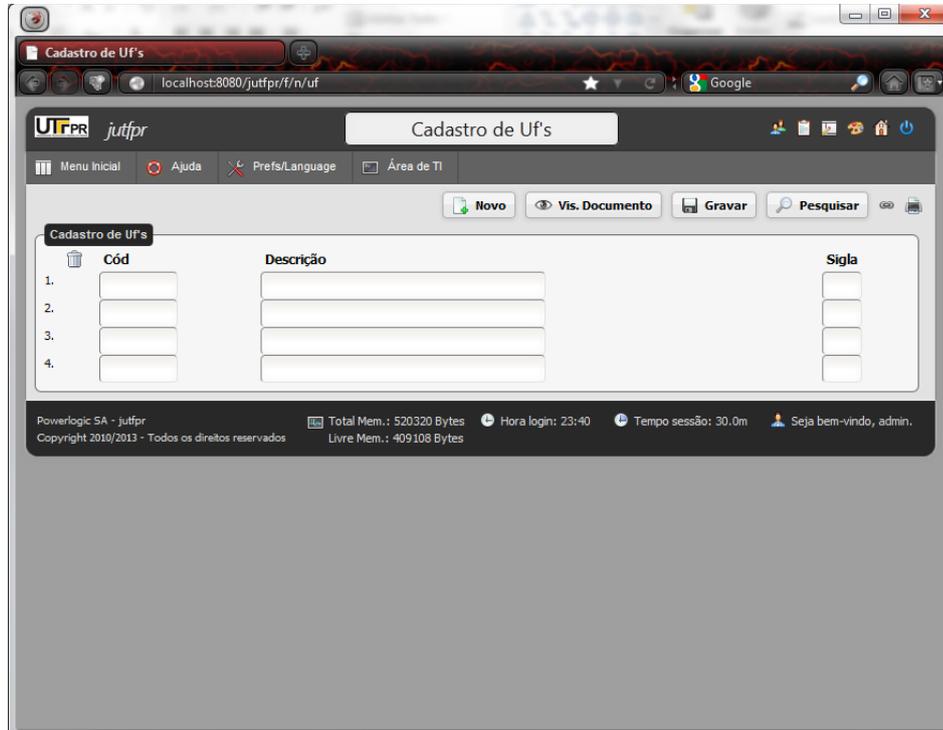


Figura 28 – Tela de Manutenção de UF's
Fonte: Autoria própria

O módulo de UF é um modulo composto por apenas uma tela que o usuário utilizará para realizar toda manutenção referente ao cadastro de UF's para posterior integração com um cadastro de endereço.

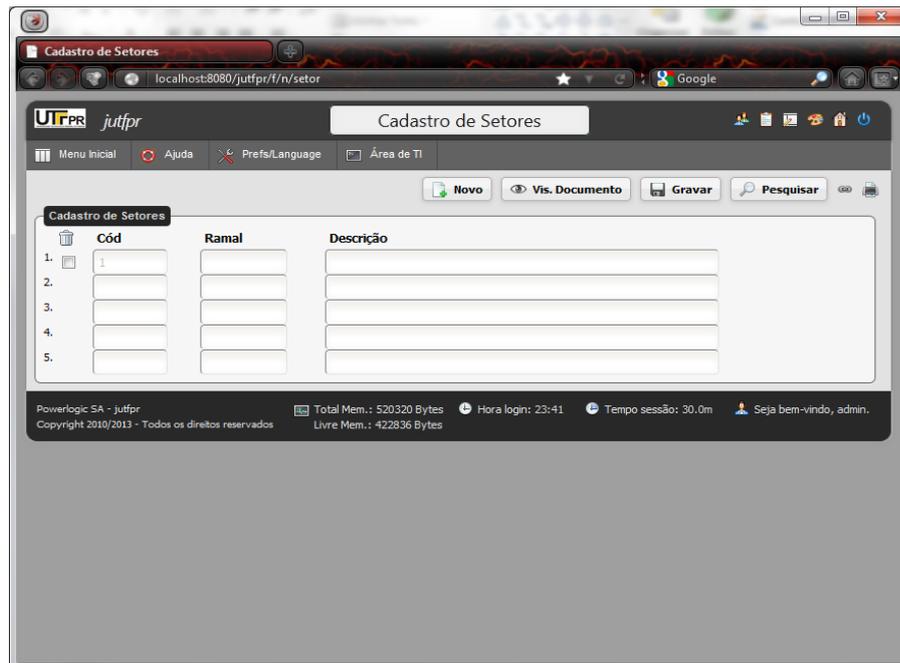


Figura 29 – Tela de manutenção de setores
Fonte: Autoria própria

O módulo de Setor é um modulo composto por apenas uma tela que o usuário utilizará para realizar toda manutenção referente ao cadastro de setores para posteriormente integrar com cadastros de cargos e funcionários.

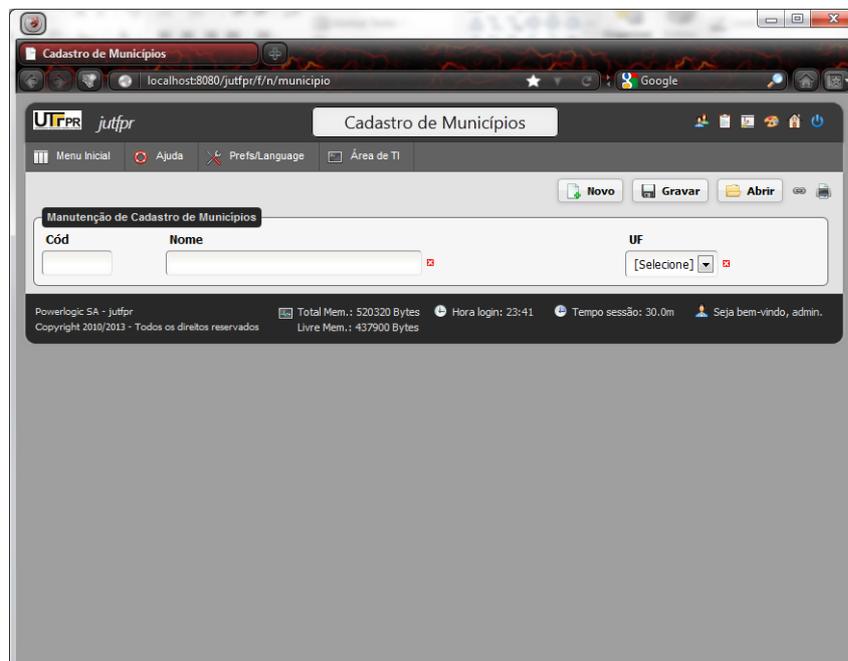


Figura 30 – Tela de manutenção de municípios
Fonte: Autoria própria

O módulo de Municípios é composto por apenas uma tela que o usuário utilizará para realizar toda manutenção referente ao cadastro de municípios. A partir desse módulo é criado o primeiro vínculo com o cadastro de UF's.

Figura 31 – Tela de manutenção de cargos
Fonte: Autoria própria

O Módulo de Cargo permite gerenciar todos os Cargos de uma empresa, vinculando-os a setores e criando regras de pagamento de acordo com a política interna.

5.2 CONSIDERAÇÕES SOBRE O DESENVOLVIMENTO DA APLICAÇÃO

Finalizando a construção e descrição dos métodos utilizados para construção da ferramenta com o *framework* Jaguar, concluiu-se que a utilização do *framework* foi de grande auxílio devido a otimização dos processos envolvidos além da interface do *framework* que de uma forma intuitiva facilita a operação da ferramenta sem a necessidade constante de recorrer a manuais e a documentação de ajuda do *framework*.

Como otimização foram consideradas principalmente a capacidade que o *framework* possui para a criação do banco de dados que se dá de forma intuitiva a

partir dos campos presentes em cada formulário. Também considerando os formulários de entrada de informações, as tarefas de CRUD também foram automatizadas pelo Jaguar, apenas referenciando as propriedades de cada campo o *framework* se encarrega de adicionar as opções de criação, edição e exclusão.

O tempo de criação das páginas e de sua concepção visual também teve auxílio do *framework*, onde foi possível escolher entre vários arquivos chamados “*templates*” que já trazem o visual de tabelas, botões, barras e itens de navegação em geral ou da criação de um *template* pessoal. No projeto em questão um dos *templates* já existente no Jaguar foi adaptado visualmente de acordo com a escolha da equipe de desenvolvimento.

O tempo estimado para a criação de cada classe e suas camadas foi de sete a quinze dias de acordo com sua complexidade e quantidade de campos, durante o trabalho e familiarização com o uso do *framework* muitas das tarefas tiveram sua conclusão antes do período previamente determinado.

6 CONCLUSÃO E TRABALHOS FUTUROS

Este capítulo apresenta a conclusão sobre os resultados obtidos na combinação entre Scrum e Jaguar no desenvolvimento da ferramenta de auxílio à implantação de planos de cargos e salários, bem como possíveis trabalhos futuros para aperfeiçoamento da ferramenta.

6.1 CONCLUSÃO

Do objetivo inicial de avaliar o desenvolvimento conjunto do *framework* Jaguar com o desenvolvimento ágil proporcionado pelo Scrum na criação de uma ferramenta de auxílio na implantação de planos de cargos e salários foi possível observar uma importante contribuição de ambas as metodologias.

No que diz respeito aos planos de cargos e salários, foram encontrados diferentes métodos para levantamento de dados e informações, além de inúmeras propostas de como realizar a tarefa em diferentes empresas variando de quantidade de funcionários, área de atuação, diferenças entre sua implantação no setor público e na iniciativa privada.

A conclusão por criar uma ferramenta que apenas realiza o levantamento e visualização das informações surgiu pela dificuldade de agregar todas as metodologias numa ferramenta só, mas principalmente por se tratar de uma esfera de atuação em que a tecnologia da informação pode ser de grande auxílio, mas que a decisão final não pode ser automatizada e sempre dependerá do fator humano. Como exemplo, em alguns casos a ferramenta pode apresentar todos os dados favoráveis à promoção de determinado funcionário, mas sua chefia imediata, devido a questões pessoais pode acabar negando essa promoção. Dados os fatos, a ferramenta é apresentada como um sistema que auxilia a decisão, mas que não pode decidir pelo cliente.

O *framework* Jaguar agrega diversas outras ferramentas para as mais diversas tarefas de desenvolvimento, o que evitou grande parte do retrabalho nas tarefas de padronização de interface, onde a interface base foi criada no padrão de *templates*, ou seja, cria-se um modelo que se replica em todas as outras telas. As

tarefas de CRUD também foram automatizadas, apenas referenciando campos e suas atribuições, automatizando também a criação do banco de dados.

A metodologia Scrum, colaborou para integrar a equipe de trabalho, estabelecendo prazos reais, atribuições individuais e coletivas, bem como as especificações do projeto e levantamento de requisitos.

6.2 TRABALHOS FUTUROS

Alguns fatores que influenciam na criação de um plano de cargos e salários podem se caracterizar como melhorias ao sistema, como por exemplo, a avaliação de desempenho, que tem como objetivo analisar cada funcionário a partir da sua própria visão na empresa, na visão de sua chefia imediata e da sua equipe de trabalho e através de dados quantitativos como, por exemplo, um sistema de pontuação, que pode influenciar diretamente na remuneração deste funcionário independente de seu cargo ou tempo de empresa.

No que diz respeito ao trabalho com *frameworks*, podem ser realizados testes comparativos entre o Jaguar e outros frameworks para determinar sua eficiência em relação as demais ferramentas com mesmo propósito na esfera de desenvolvimento de software.

A ferramenta foi criada como um sistema de apoio à decisão passiva, ou seja, não atuando diretamente nas decisões a serem tomadas sobre seus resultados, pois ainda depende só e exclusivamente do ser humano para decidir como utilizar os dados levantados, como trabalho futuro pode ser apresentada a ideia de criar um sistema de apoio à decisão ativo, utilizando de fórmulas estatísticas e inteligência artificial para uma solução mais concreta de auxílio nas tomadas de decisões, mesmo que, ainda assim, a decisão final envolverá sempre o fator humano.

REFERÊNCIAS

BERGE, Sandro T. **Estrutura de cargos na administração pública municipal.** Caxias do Sul. Educs. 2005.

CHIAVENATO, Idalberto. **Gerenciando com as pessoas.** 1 ed. Campus. 2005.

CISNEIROS, H. **Modelo de desenvolvimento ágil Scrum**, Mai. de 2009. Disponível em: <<http://www.devin.com.br/modelo-scrum/>> Acesso em 2 abr. 2012.

Diário Oficial da União, Nº 13, 19/01/2011, Seção 1. Disponível em: <http://convergenciadigital.uol.com.br/inf/in_softwarelivre.pdf> Acesso em 19 abr. 2012.

DUARTE, Anna Carolina L. **Afinal, o que é o JAGUAR?** Disponível em: <<http://empauta.assespro-mg.org.br/?p=5189>> Acesso em 01 mar. 2012.

FAYAD, M. E., SCHMIDT, D. C. **Object-oriented Application frameworks.** Communications of the ACM, v. 40, 10 p., 1997.

MATOS, F.G. **Fator QF- Ciclo de felicidade no Trabalho.** São Paulo: Makron Books, 1997.

MIRANDA, Karina , MIRANDA Elenir C. F. **Compreendendo a gestão de pessoas.** Disponível em: <http://artigos.netsaber.com.br/resumo_artigo_16677/artigo_sobre_compreendendo_a_gestao_de_pessoas> Acesso em 19 abr. 2012.

MORAES, Lucas. **A vida com Scrum** Disponível em: <<http://alemdati.wordpress.com/2010/03/26/a-vida-com-scrum/>> Acesso em 19 abr. 2012.

POWERLOGIC. **Documentação do Framework Jaguar: Módulo B - Casos de Uso Centrados em Dados (Primários).** Disponível em: <<http://www.powerlogic.org/documentacao.html>> Acesso em: 14 jan. 2012.

PRESSMAN, Roger S. **Engenharia de Software**. 6 ed. São Paulo: McGraw-Hill, 2006.

SCHWABER Ken, SUTHERLAND Jeff. **Guia do Scrum**. Disponível em:
<<http://www.scrum.org/storage/scrumguides/Scrum%20Guide%20-%20Portuguese%20BR.pdf>> Acesso em 19 abr. 2012.

TACHIZAWA, Takeshy; FERREIRA, Victor Cláudio Paradela; FORTUNA, Antônio Alfredo Melo. **Gestão com pessoas. Uma abordagem aplicada às estratégias de negócios**. 5 ed. Rio de Janeiro: FGV, 2006.

TAKEUCHI, Hirotaka; NONAKA Ikujiro. **The New Product Development Game**. Disponível em:
<<http://www.sao.corvallis.or.us/drupal/files/The%20New%20New%20Product%20Development%20Game.pdf>> Acesso em 19 abr. 2012.

ZIMPECK, B G. **Administração de salários**. 7 ed. São Paulo: Atlas, 1990.