

**CENTRO UNIVERSITÁRIO DE ANÁPOLIS UNIEVANGÉLICA
ENGENHARIA DE COMPUTAÇÃO**

**DANIEL FERREIRA RABELO
MARCO VINICIUS ISECKE CÂNDIDO**

**ANÁLISE DE DESEMPENHO DE BANCO DE DADOS NOSQL EM UM
SISTEMA QUE UTILIZA UM BANCO DE DADOS RELACIONAL E
NÃO RELACIONAL PARA ARMAZENAMENTO DE DADOS**

TRABALHO DE CONCLUSÃO DE CURSO

ANÁPOLIS - GO

2017

DANIEL FERREIRA RABELO
MARCO VINICIUS ISECKE CÂNDIDO

**ANÁLISE DE DESEMPENHO DE UM SISTEMA QUE UTILIZA UM
BANCO DE DADOS RELACIONAL E NÃO RELACIONAL PARA
ARMAZENAMENTO DE DADOS**

Trabalho de Conclusão de Curso
apresentada como requisito parcial à
obtenção do título de Bacharel em
Engenharia de Computação, do curso de
Engenharia de Computação, do Centro
Universitário de Anápolis
UniEVANGELICA

Orientadora: Aline Dayany de Lemos

ANÁPOLIS - GO

2017

Aos meus pais, que não mediram esforços para que eu chegasse até esta etapa da minha vida.

Daniel Ferreira Rabelo

Dedico à minha Vó Freida (*in memoriam*) a minha Mãe Adriana, e a meu querido amigo Rodrigo, por acreditarem, me apoiarem, e me guiarem com toda a destreza e sabedoria de que eu precisava.

Marco Vinicius Isecke Candido

AGRADECIMENTOS

Agradeço à minha família, principalmente meus pais Elias Rabelo e Suzana de Melo Ferreira Rabelo, e irmãos Danillo Ferreira Rabelo e Murillo Ferreira Rabelo por todo apoio e incentivo. A Ana Lúcia Mascarenhas e Mozart Mascarenhas, por cederem à sua casa para mim neste período, obrigado! Aos amigos, em especial aqueles cuja amizade germinou e se fortaleceu durante os anos de curso, Danielly de Souza e Silva, Caroline Paulina Jacob, Gustavo Henrique Silva, Igor Nathan Valdivino Vieira, Larah Roberta Gonçalves, Luis Filipe Sales Messias, Marco Vinicius Isecke Candido e Yask Carolinne Pereira de Faria.

Minha gratidão ao Centro Universitário de Anápolis UniEvangélica e seu corpo docente, com menção especial à minha orientadora Aline Dayany de Lemos, pela paciência e revisão deste trabalho, e aos professores Viviane Carla Batista Pocivi e Diego de Moura Silva, pelas portas abertas, tornando possível o meu ingresso ao mercado de trabalho. Agradeço à Deus, pela força e coragem durante toda esta longa caminhada.

Por fim, meu mais sincero obrigado a quem, de forma direta ou indireta, ajudou com este trabalho e com minha formação profissional.

Daniel Ferreira Rabelo

AGRADECIMENTOS

Agradeço em primeiro lugar à minha tia e avó (*in memoriam*) que fizeram de tudo para me tornar o que sou, acreditaram no meu sonho e me ajudaram à conquista-lo, sem elas eu não seria ninguém.

Agradeço também aos meus grandes e bons amigos, que fiz e que perduram comigo durante esses cinco anos, saibam que vocês souberam me amparar em momentos difíceis e me alegraram mais ainda em momentos felizes. Aos que conheci nessa jornada, e que espero que fiquem ao meu lado por muito mais tempo, Daniel Ferreira Rabelo, Gustavo Henrique Silva, Larah Roberta Gonçalves, Igor Nathan Valdivino Vieira, Evelyn Tondato, Larissa Oliveira, Osilmar Mendonça Filho. Aos que me aguentaram a mais tempo, e sempre estão juntos a mim: Marina da Silva Montelo, Marília Raquel, Ludemeula Fernandes. E ao meu irmão Luis Felipe Isecke, e irmãos de coração, Fernando Medina e Raphael Laranjeira.

E em especial, a uma pessoa que esteve comigo esses anos, e está me aguentando de forma especial, e me dando apoio, amor e carinho, Luis Filipe Sales Messias.

Agradeço por último aos meus queridos mestres, que nos guiaram e aguentaram de forma que eu colocaria como, brilhante!

Marco Vinicius Isecke Cândido

“A persistência é o caminho do êxito”
(Charles Chaplin)

RESUMO

Por meio do avanço das tecnologias voltadas a escalabilidade de um banco de dados, o uso de bancos voltados a performance que administram uma massiva quantidade de dados (*big data*) tornou-se mais acessível. Esses bancos são conhecidos como bancos *NoSQL*, e são um novo paradigma computacional. Entretanto, o uso de *NoSQL* associado à tecnologia *big data* não é uma regra. Este trabalho, pretende desmitificar através do desenvolvimento de um sistema que irá armazenar poucos dados em seu banco de dados não relacional, que para se utilizar *NoSQL* é necessário possuir um Big Data, além disso, será analisado qual o desempenho do banco de dados neste cenário. Essa análise representam a contribuição desse trabalho, provendo uma análise comparativa teórica de forma que também possibilitará a realização de futuras implementações práticas de *software* de baixo volume de dados baseado nos dados levantados e apresentados neste trabalho.

Palavras-chave: Banco de Dados. *NoSQL*. BigData. *Software*. Banco de Dados Relacional. Banco de Dados Não-Relacional. *Java*. *MongoDB*. *PostgreSQL* .

LISTA DE ILUSTRAÇÕES

Figura 1 – Injeção de Dependência.....	28
Figura 2 – Diagrama de Caso de Uso SDC.....	33
Figura 10 – Ciclo de vida <i>SCRUM</i>	36
Figura 11 – Perspectiva do Openup.....	37
Figura 12 – Disposição do <i>Openup</i>	38
Figura 13 – Processo de Desenvolvimento de <i>Software</i>	39
Figura 14 – Arquitetura do <i>software</i>	41
Figura 15 – Tempo de resposta.....	46
Figura 16 – <i>Checklist</i> de Desempenho do Banco de Dados.....	48
Figura 17 – Herança.....	49
Figura 18 – Abstração e encapsulamento.....	50
Gráfico 1 – Demonstrações Contábeis.....	32

LISTA DE TABELAS

Tabela 2 – Importar Plano de Contas.....	52
Tabela 3 – Visualizar Plano de Contas	52
Tabela 4 – Importar Balancete	52
Tabela 5 – Gerar relatório de balanço patrimonial	53
Tabela 6 – Gerar relatório de demonstração de fluxo de caixa	53
Tabela 7 – Gerar relatório de demonstração das mutações do patrimônio líquido ...	53
Tabela 7 – Resultado do Checklist de Análise de Desempenho	55

SUMÁRIO

1 INTRODUÇÃO	13
1.1 PROBLEMA	13
1.2 JUSTIFICATIVA	13
1.3 OBJETIVOS	14
1.3.1 Objetivo Geral	14
1.3.2 Objetivos Específicos	15
1.4 METODOLOGIA	15
2 REFERENCIAL TEÓRICO	17
2.1 BANCO DE DADOS RELACIONAIS	18
2.1.1 MySQL	19
2.1.2 SQLServer	20
2.1.3 PostegreSQL	20
2.1.4 Oracle	21
2.2 BANCO DE DADOS NÃO-RELACIONAIS	21
2.2.1 Classificação de Bancos de Dados Não-Relacionais	22
2.2.1.1 Banco de dados de chave-valor	22
2.2.1.2 Banco de dados de documentos	23
2.2.1.3 Armazenamento em famílias de colunas	23
2.2.1.4 Banco de dados de grafos	23
2.3 PROCESSO DE DESENVOLVIMENTO DE <i>SOFTWARE</i>	24
2.3.1 <i>Framework Scrum</i>	24
2.3.2 <i>OpenUp</i>	25
2.4 ARQUITETURA	26
2.4.1 JSF	27
2.4.2 CDI	28
2.4.3 MANAGED BEANS	28
2.4.4 JPA	29
3 DESENVOLVIMENTO	30
3.1 SISTEMA DE DEMONSTRAÇÕES CONTÁBEIS - SDC	30
3.1.1 VISÃO DO PRODUTO	33
3.1.2 ESTÓRIAS DE USUÁRIO	33
3.2 PROCESSO DE DESENVOLVIMENTO DE <i>SOFTWARE</i>	35
3.3 <i>SCRUM</i>	35
3.4 <i>OPENUP</i>	37
3.4.1 PROCESSO DE DESENVOLVIMENTO HÍBRIDO – <i>SCRUM + OPENUP</i>	38
3.5 ARQUITETURA DO <i>SOFTWARE</i>	41
3.6 BENEFÍCIOS E MALEFÍCIOS DO BANCO DE DADOS NÃO RELACIONAL ..	43

3.7 MÉTRICAS DE DESEMPENHO DE BANCO DE DADOS NÃO-RELACIONAL	44
3.7.1 DEFINIÇÕES DAS MÉTRICAS DE DESEMPENHO.....	45
3.7.1.1 Métricas de Velocidade.....	45
3.7.1.1.1 <i>Tempo de Resposta</i>	45
3.7.1.1.2 <i>Vazão ou Throughput</i>	46
3.7.1.1.3 <i>Utilização</i>	46
3.7.1.2 Consistência.....	47
3.7.1.3 Disponibilidade.....	47
3.8 CHECKLIST DE ANÁLISE DE DESEMPENHO DO BANCO DE DADOS.....	47
3.9 DESENVOLVIMENTO DA APLICAÇÃO.....	49
3.10 EXECUÇÃO DOS TESTES.....	51
3.10.1 AMBIENTE DE SIMULAÇÃO.....	51
3.10.2 TESTE DE CARGA.....	51
4 RESULTADOS ALCANÇADOS.....	54
ANEXO A – CARTA DE AUTORIZAÇÃO.....	62
ANEXO B – TESTES DE CARGA	
4663	
ANEXO C – ESTÓRIAS DE USUÁRIO	
1367	

1 INTRODUÇÃO

1.1 PROBLEMA

Segundo Vieira; Figueiredo; Liberatti; Viebrantz, (2012) “A quantidade de dados gerada diariamente em vários domínios de aplicação como, por exemplo, da *Web*, redes sociais, redes de sensores, entre diversos outros, estão na ordem de algumas dezenas, ou centenas, de *terabytes*”. Nesse contexto, o movimento *NoSQL* vem ganhando força, essa nova geração de banco de dados é frequentemente utilizada a tecnologias *Big Data*, pois o *NoSQL* é comumente usado em projetos de *software* de grandes empresas no qual geram um alto volume de dados. Pensando nisso, como se comporta um sistema que armazena poucos dados (sistema que não utilize a tecnologia *Big Data*) que armazena seus dados em dois bancos de dados, um relacional e outro não relacional?

1.2 JUSTIFICATIVA

Os bancos de dados estão cada vez mais presentes em nosso dia-a-dia, visto que a maioria das atividades que realizamos envolve, direta ou indiretamente, o uso de uma base de dados. Segundo Elmasri e Navathe (2011, p.3), um banco de dados “é uma coleção de dados relacionados”, ou seja, sempre que possuímos um grupo de informações que relacionam entre si e tratam de um mesmo assunto, temos um banco de dados.

Banco de dados relacionais hoje são a tecnologia predominante para armazenar dados estruturados. “A partir de 1970 o armazenamento de dados baseado em cálculo relacional foi bastante utilizado e muitos pensaram ser a única alternativa para o armazenamento de dados acessível por vários clientes de uma forma consistente” (FERNANDES, 2013, p.2). Porém, com a proliferação das redes sociais e a intensa interação entre usuários da *internet*, a cada dia mais e mais dados são inseridos e, recuperá-los para análise em alta velocidade é quase impossível.

O pensamento sobre como armazenar grandes quantidades de dados e como processar esses dados vem sendo bastante questionado, e isso levou ao

surgimento de uma grande variedade de soluções. O movimento, bem como as novas formas de armazenamento de dados receberam o termo de *NoSQL (Not Only SQL)*, que são bancos de dados não-relacionais.

Segundo Porcelli. (2016):

NoSQL é um movimento que promove soluções de armazenamento de dados não relacionais. Ele é composto por diversas ferramentas que, de forma particular e específica, resolvem problemas como tratamento de grandes volumes de dados, execução de consultas com baixa latência e modelos flexíveis de armazenamento de dados.

O uso de *NoSQL* se adéqua muito bem à tecnologia *Big Data*, que é o termo que descreve o imenso volume de dados – estruturados e não estruturados. Entretanto, apesar de muitos associarem *NoSQL* com *Big Data*, não existe uma regra que limita que *NoSQL* deve ser usado somente com *Big Data*.

Portanto, utilizando dos conceitos e práticas de *NoSQL* será desenvolvido um *software* que armazenará um pequeno volume de dados que utilizará um banco de dados relacional para garantir a integridade e segurança dos usuários, armazenando dados de cadastro de usuários e autenticação de usuário, e um banco de dados não relacional para armazenar os dados gerados decorrentes às funcionalidades de emissão de relatórios do *software*. A partir deste *software*, será analisado o desempenho do banco de dados, afim de verificar se é viável utilizar um *NoSQL* em sistema que não utiliza a tecnologia *Big Data*.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Desenvolver e analisar o desempenho de um sistema que tem um baixo armazenamento de dados (não seja *BigData*) que utilize de um banco de dados relacional e não relacional (*NoSQL*) para armazenar suas informações, afim de investigar se o uso de *NoSQL* seria uma solução adequada para sistemas que estejam imersos neste cenário de armazenamento de dados.

1.3.2 Objetivos Específicos

- Definir o modelo de bancos de dados não relacionais (bancos de dados de chave-valor, bancos de dados de documentos, armazenamento em famílias de colunas e banco de dados de grafos);
- Definir arquitetura do sistema;
- Definir os requisitos funcionais do sistema;
- Realizar a implementação dos requisitos funcionais do sistema;
- Traçar os benefícios da utilização de banco de dados não relacional a partir de fundamentação teórica;
- Traçar os malefícios da utilização de banco de dados não relacional a partir de fundamentação teórica;
- Traçar métricas de desempenho de um banco de dados não relacional em um sistema que armazena poucos dados;
- Elaborar *checklist* de avaliação de desempenho de um banco de dados não relacional em um sistema que armazena poucos dados;
- Realizar testes no banco de dados, aplicando o *checklist desenvolvido*, afim de analisar o desempenho do mesmo;
- Analisar os dados obtidos;
- Elaborar a conclusão da análise, afim de comprovar a viabilidade ou não da utilização de bancos de dados não relacionais em sistema com pouco armazenamento de dados.

1.4 METODOLOGIA

A pesquisa quantitativa foi realizada a partir de um *software* desenvolvido pelos autores. O *software* tem como objetivo emitir relatórios de demonstrações contábeis de diversos segmentos de entidades empresariais. Os dados destes relatórios são provenientes do balancete e plano de contas da entidade empresarial que será importado para o banco de dados do *software*. O *software* foi desenvolvido utilizando como base a metodologia de desenvolvimento de *software* Ágil que mescla *Scrum* e *OpenUP*. Diariamente, por cerca de 4 (quatro) horas os autores pretendem trabalhar no desenvolvimento das funcionalidades do sistema. A

linguagem de programação selecionada é *Java Web*, tecnologia de banco de dados relacional será o *PostgreSQL* e não-relacional, *MongoDB*.

A pesquisa se resume em realizar testes de desempenho no banco de dados não relacional, afim de analisar se o uso de banco de dados não relacional é viável para sistemas que armazenam poucas informações em seu banco de dados. Para realização dos testes de desempenho de banco de dados, foram utilizadas ferramentas de mensuração de desempenho. Nessas ferramentas, foi definidas cargas de processamento de transações que simulam as funcionalidades do sistema desenvolvido pelos atores onde foi verificado a amplitude na medição do desempenho.

O instrumento de coleta de dados é um *checklist*, que mensurou o desempenho do sistema em relação ao seu banco de dados. Para elaboração deste *checklist*, foi definido os testes de desempenho que serão utilizados em cima de conceitos referentes à banco de dados não relacional, onde foi levantado os principais pontos de importância para análise de desempenho do banco de dados.

Em sequência a pesquisa foi concluída, após a análise dos dados de desempenho do banco de dados levantados, comparando com as principais características de um banco de dados não relacional e o seu desempenho, afim de verificar que a utilização de banco de dados não-relacional em sistemas que não utilizem a tecnologia *Big Data* é viável.

2 REFERENCIAL TEÓRICO

Segundo Elmasri e Navathe (2011, p. 2), “os bancos de dados e os sistemas de bancos de dados se tornaram componentes essenciais no cotidiano da sociedade moderna. No decorrer do dia, a maioria de nós se depara com atividades que envolvem alguma interação com os bancos de dados”.

A estabilidade dessa área é de grande valor. Os dados de uma organização perduram muito mais tempo do que seus programas. É importante termos um armazenamento de dados estável, que seja bem compreendido e que possa ser acessado a partir de muitas plataformas de programação de aplicativos, (SADALAGE, FOWELER, 2013, p.13).

Date, (2003, p.10) diz que, “um banco de dados é uma coleção de dados persistentes, usada pelos sistemas de aplicação de uma determinada empresa”. Em outras palavras, um banco de dados é um local onde são armazenados dados necessários ao suporte das atividades de determinada entidade, sendo este diretório a fonte de dados para os *softwares* atuais e os que vierem a existir.

O crescimento do armazenamento virtual se dá de forma exponencial, após os anos 2000, preocupando vários desenvolvedores por conta da falta de espaço em armazenamento. Dados da *IBM* apontam que em 2015 “diariamente são criados 4,5 quintilhões de *bytes* de dados a partir de uma variedade de fontes de informação”, (OLIVEIRA, 2015). Este fato é decorrente da, cada vez mais frequente, conexão de empresas à internet, como redes sociais, companhias de telefonia, serviços de *streaming*, dentre outras.

O paradigma sobre o armazenamento de dados em aplicações, trazem à tona estudos sobre as características necessárias em modelos de banco de dados para que esse crescente volume de informações seja gerenciado de forma adequada, (OLIVEIRA, 2015).

Oliveira (2015), *et al*, diz que:

Um modelo de banco de dados é uma descrição dos tipos de informações que estão armazenados em um banco de dados. Um dos modelos de banco

de dados amplamente usados até os dias de hoje é o modelo relacional. Porém, constatou-se que os modelos de banco de dados relacionais apresentam limitações ao trabalhar com grandes volumes de dados.

A difusão do tema “modelos de bancos de dados não-relacionais” provém desse aumento do volume de dados gerados na web, e da percepção de que o modelo relacional pode se mostrar ineficiente quando utilizado para administrar grande quantidade de dados, (OLIVEIRA, 2014).

Sadalage, e Fowler (2013, p.34) afirmam em seu livro que “o termo *NoSQL* tenha feito sua primeira aparição no final da década de 1990 com o nome de um banco de dados relacional de código aberto (*open source*)”.

Em 2006, o artigo “BigTable: The Distributed Storage System for Structured Data” publicado pelo Google resgatou o termo *NoSQL* como um conceito de gerenciamento de megadados. O *BigTable* foi descrito como um banco extremamente escalável e tolerante a falhas onde os dados inseridos já entram indexados, tornando assim mais rápida a consulta aos dados armazenados (OLIVEIRA, 2015).

O grande incentivo para o movimento *NoSQL* é resolver o problema de escalabilidade dos bancos tradicionais, sabendo que pode ser muito caro e/ou complexo escalar um banco de dados relacional. Motivada nesse novo conceito disseminado, a comunidade de *software* livre e código aberto em geral desenvolveram diversas soluções de bancos de dados não-relacionais seguindo diferentes vertentes, (STEPPAT. 2009).

2.1 BANCO DE DADOS RELACIONAIS

O modelo relacional em banco de dados é fundamentado no princípio de que dados são guardados em tabelas. Segundo Elmasri e Navathe (2011, p.39), “o modelo relacional representa o banco de dados como uma coleção de relações. Informalmente, cada relação é semelhante a uma tabela de valores”.

Esse modelo ainda é bastante utilizado pelo fato de prover acesso facilitado aos dados, possibilitando aos usuários utilizar uma grande variedade de abordagens no tratamento das informações, além da possibilidade de uso dos

sistemas gerenciadores de banco de dados, que executam comandos na linguagem SQL (*Structured Query Language*) e têm a responsabilidade de administrar o acesso, a manipular e a organizar os dados, principalmente no que diz respeito à segurança, (OLIVEIRA, 2015).

Com o passar do tempo e com o crescente volume de dados gerados a partir da expansão virtual, identificou-se que o modelo relacional não é tão escalável quanto necessário. Quando utilizado para gerenciar um grande volume de informações e cargas de trabalhos típicas de operações modernas de grande carga, incluindo o dimensionamento de conjunto de dados, o banco de dados relacional perdeu sua performance consideravelmente. Neste sentido, quanto mais dados forem gerados, mais recursos de hardware serão necessários, como memórias e discos, para que a qualidade do serviço seja mantida, (OLIVEIRA, 2016, P.187).

Os SGBDs, ou sistemas de gerenciamento de banco de dados, são *softwares* que permitem criar bases de dados, permitindo ainda realizar ações de modificações, exclusões e inserções de dados na base de dados criada. Entre os SGBDs relacionais, destacam-se: *MySQL*, *SQLServer*, *PostgreSQL* e *Oracle*.

2.1.1 MySQL

Segundo o site da Oracle, o “*MySQL* é o banco de dados de código aberto mais conhecido do mundo. Com comprovado desempenho, confiabilidade e facilidade de uso”. Entretanto, o site *Abcarticulos*, no artigo “vantagens e desvantagens do *MySQL*” publicado em 2016, diz que “vulnerabilidades de segurança podem surgir porque nenhum padrão de criptografia pode ser usado dentro de um procedimento armazenado no banco de dados”. No mesmo artigo, apontam ainda como desvantagem “o tamanho da base de dados. Não pode trabalhar eficientemente com grandes bancos de dados”, por essas desvantagens o *MySQL* não foi selecionado para compor a arquitetura do sistema.

2.1.2 SQLServer

O Microsoft *SQLServer* é um SGBD utilizado em pequenas e grandes empresas. O SGBD apresenta várias vantagens em relação à gerenciamento de banco de dados, no entanto, o *software* possui um alto custo financeiro para aquisição, inviabilizando a utilização do mesmo. (MICROSOFT, 2017).

2.1.3 PostgreSQL

O *PostgreSQL* é um poderoso sistema gerenciador de banco de dados relacional de código aberto e gratuito. Possui uma arquitetura que comprovadamente ganhou forte reputação de confiabilidade, integridade de dados e conformidade a padrões. “Oferece aos usuários processos de validação, verificação e garantias de integridade dos dados, controle de concorrência, recuperação de falhas, segurança, controle de transações, otimização de consultas, dentre outros” (BRITO, 2010, P.1).

O *PostgreSQL* é um sistema gerenciador de banco de dados objeto-relacional (SGBDOR), baseado no *Postgres* versão 4.2 desenvolvido pelo Departamento de Ciência da Computação da Universidade da *Califórnia* em *Berkeley* (OLIVEIRA, 2007, p.14).

Furlan, (2015) afirma que,

O *PostgreSQL* é um descendente de código fonte aberto deste código original de Berkeley, que suporta grande parte do padrão *SQL* e oferece muitas funcionalidades modernas, como: Comandos complexos, Chaves estrangeiras, Gatilhos, Visões, Integridade transacional e Controle de simultaneidade multiversão.

Além disso, o *PostgreSQL* pode ser ampliado pelo usuário de muitas maneiras como, por exemplo, adicionando novos: Tipos de dado, Funções, Operadores, Funções de agregação, Métodos de índice e Linguagens procedurais.

“Devido à sua licença liberal, o *PostgreSQL* pode ser utilizado, modificado e distribuído por qualquer pessoa para qualquer finalidade, seja particular, comercial ou acadêmica, livre de encargos”, (OLIVEIRA, 2007, p.14).

Quanto mais dados são inseridos nos bancos relacionais, a performance do banco diminui. Para atenuar esta situação necessita-se aumentar o processamento. Existem a escalabilidade vertical – mais recursos para a máquina, e a escalabilidade horizontal – dividir o processamento em máquinas. (COELHO, 2004)

2.1.4 Oracle

Oracle é um SGBD editado pela Oracle Corporation, líder mundial desta área. Ela foi criada em 1977 por Lawrence Ellison, Bob Miner e Ed Oates sob o nome Relational Software Incorporated (RSI) e comercializa um sistema de gestão de bancos de dados relacionais (SGBDR ou RDBMS) chamado Oracle.

O Oracle se divide em duas versões: Oracle Server Standard (Padrão), uma versão que compreende as ferramentas mais comuns da solução Oracle, e Oracle Server Enterprise Edition, Segundo o site da Oracle, “ele garante a definição e a manipulação dos dados, a coerência dos dados, a confidencialidade dos dados, a integridade dos dados, o backup e a restauração dos dados e o acesso aos dados”.

2.2 BANCO DE DADOS NÃO-RELACIONAIS

Para evitar a escalabilidade em banco de dados relacionais, a elaboração de bancos de dados não relacional, deu-se início:

Capazes de gerenciar dados semiestruturados oriundos de diversas origens e que possibilitavam escalabilidade mais barata e menos complexa, não necessitando de servidores muito robustos e nem um grande número de profissionais para o administrar (OLIVEIRA, 2014).

De acordo com Moura e Casanova (1999, p.14):

A criação de sistemas de gerenciamento de banco de dados contribui de forma significativa para o aumento da produtividade em desenvolvimento de aplicações, um fator importante desde longa data. De fato, tais sistemas simplificam a tarefa de se definir aplicações que requerem o compartilhamento de informação entre usuários, programas ou organizações onde os usuários da informação, ou mesmo as fontes de informação, estão geograficamente dispersas.

“Dessa forma, os bancos de dados não-relacionais ficaram muito populares entre as grandes empresas geradoras de conteúdo, e são amplamente difundidos na comunidade *open source* e *software livre*” (OLIVEIRA, 2015).

2.2.1 Classificação de Bancos de Dados Não-Relacionais

Os bancos de dados não relacionais são classificados em bancos de dados de chave-valor, banco de dados de documentos, armazenamento em famílias de colunas e banco de dados de grafos. (SADALAGE, FOWELER, 2015)

2.2.1.1 Banco de dados de chave-valor

Segundo Sadalage e Fowler(2015):

Depósitos de chave-valor são os depósitos de dados *NoSQL* mais simples de utilizar a partir da perspectiva de uma API. O cliente pode obter o valor para uma determinada chave, inserir um valor para uma determinada chave ou apagar uma chave do depósito de dados. O valor é um *blob* que o depósito de dados apenas armazena, sem se preocupar ou saber o que há dentro dele; é responsabilidade do aplicativo entender o que foi armazenado. Já que depósitos de chave-valor sempre fazem o acesso pela chave primária, eles têm, geralmente, um ótimo desempenho e podem ser escaláveis facilmente.

2.2.1.2 Banco de dados de documentos

Documentos são o conceito principal em bancos de dados de documentos. O banco de dados armazena e recupera documentos, os quais podem ser XML, JSON, BSON, entre outros. Esses documentos são estruturas de dados na forma de árvores hierárquicas e auto descritivas, constituídas de mapas, coleções e valores escalares. Os documentos armazenados são semelhantes entre si, mas não têm de ser exatamente os mesmos (SADALAGE, FOWLER, 2015, P.133).

2.2.1.3 Armazenamento em famílias de colunas

Bancos de dados de famílias de colunas armazenam dados em famílias de colunas como linhas que tenham muitas colunas associadas, fazendo uso de uma chave de linha. Famílias de colunas são grupos de dados relacionados que, frequentemente, são acessados juntos. Por exemplo, muitas vezes acessamos as informações de perfil de um cliente ao mesmo tempo, mas não seus pedidos (SADALAGE, FOWLER, 2015, p.147).

2.2.1.4 Banco de dados de grafos

Bancos de dados de grafos permitem que você armazene entidades e também relacionamentos entre essas entidades. Entidades também são conhecidas como nodos, os quais possuem propriedades. Pense em um nodo como uma instância de um objeto do aplicativo. Os relacionamentos são conhecidos como arestas que podem ter propriedades. As arestas têm significância direcional; nodos são organizados por relacionamentos, os quais permitem que você encontre padrões interessantes entre eles. A organização do grafo permite que os dados sejam armazenados uma vez e depois interpretados de formas diferentes baseadas em relacionamentos (SADALAGE, FOWLER, 2015, p.161).

2.3 PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

Segundo Sommerville (2011, p.18), “um processo de desenvolvimento de *software* é um conjunto de atividades relacionadas que levam à produção de um produto de *software*. Essas atividades podem envolver o desenvolvimento de *software* a partir do zero em uma linguagem padrão de programação como *Java* ou *C*.”

Os processos de *software*, às vezes, são categorizados como dirigidos a planos ou processos ágeis. Abordaremos o processo ágil no desenvolvimento deste trabalho. Em processos ágeis, o planejamento é gradativo, e é mais fácil alterar o processo de maneira a refletir as necessidades de mudanças dos clientes (SOMMERVILLE, 2011, p.42).

Sommerville (2011, p.38), ainda afirma em seu livro que “os processos de desenvolvimento rápido de *software* são concebidos para produzir, rapidamente, *softwares* úteis. O *software* não é desenvolvido como uma única unidade, mas como uma série de incrementos – cada incremento inclui uma nova funcionalidade do sistema”.

2.3.1 Framework Scrum

Schwabere e Sutherland (2013, p.3), descrevem o *Scrum* no Guia do *Scrum* como sendo, “[...] 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*, “não é um processo ou uma técnica para construir produtos; em vez disso, é um *framework* dentro do qual você pode empregar vários processos ou técnicas” (SCHWABER, SUTHERLAND, 2013, p.3).

Segundo Schwaber e Sutherland(2013):

O time *scrum* é composto pelo *Product Owner*, o Time de Desenvolvimento e o *Scrum Master*. Times *Scrum* são auto-organizáveis e multifuncionais. Times auto-organizáveis escolhem qual a melhor forma para completarem seu trabalho, em vez de serem dirigidos por outros de fora do Time. Times multifuncionais possuem todas as competências necessárias para

completar o trabalho sem depender de outros que não fazem parte da equipe. O modelo de time no *Scrum* é projetado para aperfeiçoar a flexibilidade, criatividade e produtividade.

Segundo Schwaber e Sutherland (2013, p.5), o *Scrum*, possui os seguintes papéis, são eles:

- O *Product Owner*, ou dono do produto, é o responsável por maximizar o valor do produto e do trabalho do Time de Desenvolvimento. Como isso é feito pode variar amplamente através das organizações, Times *Scrum* e indivíduos.
- O Time de Desenvolvimento consiste de profissionais que realizam o trabalho de entregar uma versão usável que potencialmente incrementa o produto “Pronto” ao final de cada *Sprint*.
- O *Scrum Master* é responsável por garantir que o *Scrum* seja entendido e aplicado. O *Scrum Master* faz isso para garantir que o Time *Scrum* adere à teoria, práticas e regras do *scrum*.

O coração do *Scrum* é a *Sprint*, um time-boxed de um mês ou menos, durante o qual um “pronto”, versão incremental potencialmente utilizável do produto, é criado. *Sprints* tem durações coerentes em todo o esforço de desenvolvimento. Uma nova *Sprint* inicia imediatamente após a conclusão da *Sprint* anterior.

As *Sprints* são compostas por uma reunião de planejamento da *Sprint*, reuniões diárias, o trabalho de desenvolvimento, uma revisão da *Sprint* e a retrospectiva da *Sprint* (SCHWABER e SUTHERLAND, 2013, p.8).

2.3.2 OpenUp

OpenUp/*Basic* é um processo de desenvolvimento de *software* de código aberto projetado para equipes pequenas e co-localizadas que querem ter uma abordagem ágil para desenvolvimento. O *Openup* é um processo iterativo que é mínimo, completo e extensível, valorizando a colaboração entre a equipe e os benefícios aos interessados ao invés da formalidade e entregáveis desnecessários (MORO, 2008).

Segundo Moro (2008), o *Openup*, possui 6 (seis) papéis definidos, são eles:

- Analista de Requisitos: A pessoa nesta responsabilidade, representa os interesses do cliente e dos usuários finais recolhendo informações dos *stakeholders* para entender o problema a ser resolvido, capturando os requisitos e definindo suas prioridades.
- Arquiteto: Este papel é responsável por definir a arquitetura do *software*, incluindo a tomada das principais decisões técnicas que orientam todo o design e a implementação do projeto.
- Desenvolvedor: É responsável por desenvolver uma parte do sistema, incluindo a construção de seu design de forma que ele atenda a arquitetura e possivelmente a prototipagem da interface de usuário, e então implementar, executar o teste de unidade e integrar os componentes que são parte da solução.
- Gerente de Projeto: Conduz o planejamento do projeto, coordena as interações com os *stakeholders* e mantém a equipe de projeto focada em alcançar os objetivos do projeto.
- *Stakeholder*: Este papel representa os grupos de interesses cujas necessidades devem ser atendidas pelo projeto. É um papel que deve ser executado por qualquer pessoa que é, ou será afetado materialmente pelo resultado do projeto.
- Testador: Este papel é responsável pelas principais atividades do esforço de teste. Estas atividades incluem identificar, definir, implementar e conduzir os testes necessários, bem como registrar os resultados dos testes e analisar os resultados.

2.4 ARQUITETURA

Segundo Garlan (2000), “Uma arquitetura de *software* envolve a descrição de elementos arquiteturais dos quais os sistemas serão construídos, interações entre esses elementos, padrões que guiam suas composições e restrições sobre estes padrões”.

Para Varoto (2002) a correlação de componentes computacionais define, por si só, uma arquitetura de *software*.

2.4.1 JSF

“O JSF é um *framework* desenvolvido pela *Sun Microsystems* junto com a *Java Community Process*, projetado para ser flexível e fácil de utilizar, o *framework* disponibiliza vários componentes de interface que encapsulam diversas funcionalidades comuns em aplicações para web” (TEIXEIRA, 2008, p.21). Entre as características do JSF:

- Permite ao desenvolvedor a criação de interfaces gráficas através de um conjunto de componentes de *interface* pré-definidos;
- Fornece um conjunto de *tags* JSP para acessar os componentes;
- Permite a reutilização de componentes das páginas;
- Associa os eventos do lado cliente com os manipuladores dos eventos do lado servidor.

O objetivo principal do JSF é tornar o desenvolvimento Web mais rápido e fácil. Permite aos desenvolvedores pensar em termos de componentes, eventos, *backing beans* ou *Managed Beans*. O JSF permite pensar em estruturas de “alto nível” e suas interações, ao invés de requisição e resposta. A complexidade do desenvolvimento Web não é visualizada pelos programadores, permitindo que os desenvolvedores se concentrem mais na lógica de negócio da sua aplicação (KITO, 2005).

Segundo Kito (2005),

O JSF é executado do lado do servidor em um container Web padrão como, por exemplo, o *Jboss*. Quando o usuário clica em um botão em uma aplicação swing2 ele vai disparar um evento que pode ser tratado diretamente no código do programa desktop. Em contraste, os navegadores Web não conhecem os componentes do JSF ou eventos, pois o *browser* só sabe interpretar HTML (*HyperText Markup Language*). Então quando um botão é clicado pelo usuário em um aplicativo Faces é gerada uma requisição que será enviada através do browser para o servidor pelo protocolo HTTP. O *Faces* é responsável por traduzir essa requisição em um

evento que será processado pelo aplicativo no servidor. É responsável também por garantir que cada elemento da interface do usuário definidas no servidor seja exibida corretamente para o navegador.

2.4.2 CDI

“O CDI é uma *API (Application Programming Interface)* para injeção de dependências e contextos. As dependências funcionam, principalmente, nomeando *beans* e os vinculando aos seus pontos de injeção pelos nomes” (GIBSON, 2016).

A anotação `@Inject` do CDI para demonstrar como é possível injetar um *bean* em outra classe. A figura 1 abaixo, retrata o cenário construído neste *software*.

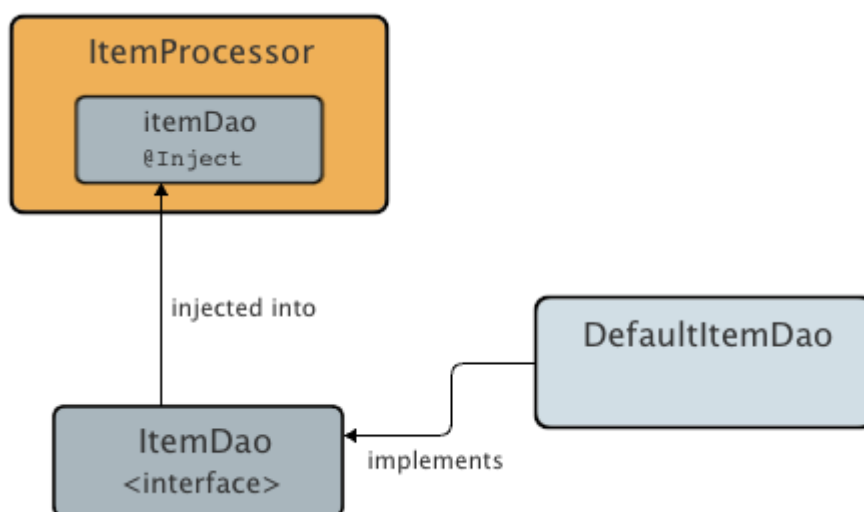


Figura 1 – Injeção de Dependência
Fonte: NetBeans

2.4.3 MANAGED BEANS

“Controladores de páginas têm como responsabilidade disponibilizar dados e métodos que possam ser referenciados por uma página *web* para entrada de dados, apresentação de dados ou tratamento de eventos. Em JSF os controladores são chamados de *Managed Beans*” (SOUSA, p.14-26).

Existem dois tipos de controladores: aqueles que não referenciam as classes de componentes no código *Java* e o que referenciam. Os primeiros são chamados *backing beans* e a referência à classe de componente é

chamada vínculo a componentes ou *Component Binding*. Os controladores de página que não referenciam componentes são chamados POJOs e apenas utilizam vínculo a dados (*Data Binding*) e vínculo a métodos (*Method Binding*). Nos controladores de página JSF a configuração do *Managed Bean* é feita usando a anotação `@ManagedBean` e alguma das anotações de escopo (SOUSA, p.14-26).

Managed Bean pode ser visto como um objeto *Java* criado, inicializado e disponibilizado para a aplicação. É preciso que o usuário saiba que o JSF faz uso da inicialização tardia do modelo, isso significa que o *bean*, em escopo específico, é criado e inicializado não quando o escopo é criado, mas sim, por demanda, ou seja, quando o *bean* é requisitado pela primeira vez (ARAÚJO, 2010, p. 151).

2.4.4 JPA

O *Java Persistence API* (JPA) é uma especificação de persistência padrão aprovada pela JCP (*Java Community Process*) em meados de 2006, como parte da especificação EJB3 (*Enterprise Java Beans 3*). O JPA lida com a forma como os dados 3 relacionais são mapeados para objetos *Java*, a maneira que esses objetos são armazenados em um banco de dados relacional para que possam ser acessados posteriormente e, a existência continuada de um estado da entidade mesmo após a aplicação que usa termina. Além de simplificar o modelo de persistência da entidade, o JPA padroniza mapeamento objeto-relacional. Neste artigo a base de dados foi mapeada utilizando do padrão JPA 2.0, para padronizar a aplicação o que torna possível que a aplicação seja independente do *framework* abordado. Sendo assim para cada teste, foi necessário somente alterar a biblioteca utilizada pela aplicação (*framework*), (JÚNIOR A. V. R. e JUNIOR, E. A. O., 2013).

3 DESENVOLVIMENTO

3.1 SISTEMA DE DEMONSTRAÇÕES CONTÁBEIS - SDC

A Contabilidade tem por objetivo contribuir para o processo decisório dos usuários através de dados financeiros suficientes para a tomada de decisões. Assim, para demonstrar a verdadeira situação de todo o grupo de empresas é utilizada as demonstrações contábeis. (PORTAL DA CONTABILIDADE, 2017)

De modo geral, de acordo com o Comitê de Pronunciamentos Contábeis 26 (CPC), as demonstrações contábeis têm o propósito de atender as necessidades informacionais de usuários.

Seu objetivo é proporcionar informação sobre a posição patrimonial e financeira, do desempenho e dos fluxos de caixa da entidade que seja útil no processo decisório ao mesmo tempo em que apresenta, de modo estruturado, a posição patrimonial e financeira e do desempenho da entidade. Assim, as demonstrações contábeis objetivam apresentar os resultados da atuação da administração na gestão da entidade e sua capacitação na prestação de contas quanto aos recursos que lhe foram confiados (LUNELLI).

Para satisfazer a esse objetivo, a entidade deve elaborar e divulgar suas informações contábeis através de um conjunto de demonstrações que deve incluir, principalmente, o Balanço Patrimonial, a Demonstração do Resultado, a Demonstração das Mutações do Patrimônio Líquido, a Demonstrações dos Fluxos de Caixa, a Demonstração do Resultado Abrangente, a Demonstração do Valor Adicionado e as Notas Explicativas. Essas informações ajudam os usuários das demonstrações contábeis na previsão dos futuros fluxos de caixa da entidade e, em particular, a época e o grau de certeza de sua geração (COMITÊ DE PRONUNCIAMENTOS CONTÁBEIS).

Segundo o site PORTAL DA CONTABILIDADE, n.d.:

- Plano de Contas Contábeis: é o conjunto de contas, previamente estabelecido, que norteia os trabalhos contábeis de registro de fatos e atos inerentes à entidade, além de servir de parâmetro para a elaboração das demonstrações contábeis;
- Balancete Contábil: é um demonstrativo auxiliar que relaciona os saldos das contas remanescentes no diário;
- Balanço Patrimonial: É a demonstração contábil destinada a evidenciar, qualitativa e quantitativamente, numa determinada data, a posição patrimonial e financeira da entidade;
- Demonstração do Resultado: Tem como objetivo principal apresentar de forma vertical resumida o resultado apurado em relação ao conjunto de operações realizadas num determinado período;
- Demonstração das Mutações do Patrimônio Líquido: Evidencia a movimentação de todas as contas do patrimônio líquido durante o exercício social, inclusive a formação e utilização das reservas não derivadas do lucro;
- Demonstrações dos Fluxos de Caixa: Demonstra quais foram as saídas e entradas de dinheiro no caixa durante o período e o resultado desse fluxo;
- Demonstração do Resultado Abrangente: Demonstra a atualização do capital próprio dos sócios, através do registro no patrimônio líquido (e não no resultado) das receitas e despesas incorridas, porém de realização financeira “incerta”, uma vez que decorrem de investimento de longo prazo, sem data prevista de resgate ou outra forma de alienação;
- Demonstração do Valor Adicionado: É o informa contábil que evidencia, de forma sintética, os valores correspondentes à formação da riqueza gerada pela empresa em determinado período e sua respectiva distribuição;
- Notas Explicativas: Visam fornecer as informações necessárias para esclarecimento da situação patrimonial, ou seja, de determinada conta, saldo ou transação, ou de valores relativos aos resultados do exercício, ou para menção de fatos que podem alterar futuramente tal situação patrimonial.

Por se tratar de informações muito técnicas é comum as entidades contratarem uma empresa de contabilidade para elaborar suas demonstrações contábeis, gerando um alto custo financeiro para a entidade, além de uma certa demora para entrega do serviço contratado.

A empresa *Gestão Consulting Services*, situada no Setor Sul da cidade de Goiânia, Goiás, Brasil, endereço Rua 119, nº 234, vide carta de autorização no anexo A, disponibilizou dados relacionados à um de seus clientes (Associação Alphaville Flamboyant Residencial).

Em fevereiro de 2017, com um total de dois funcionários trabalhando simultaneamente no período de 8hrs no cliente citado anteriormente, aponta o tempo gasto com cada relatório, conforme figura 1, em relação aos dias. Alguns relatórios são gerados concomitantemente, obedecendo a ordem de dependência entre eles.

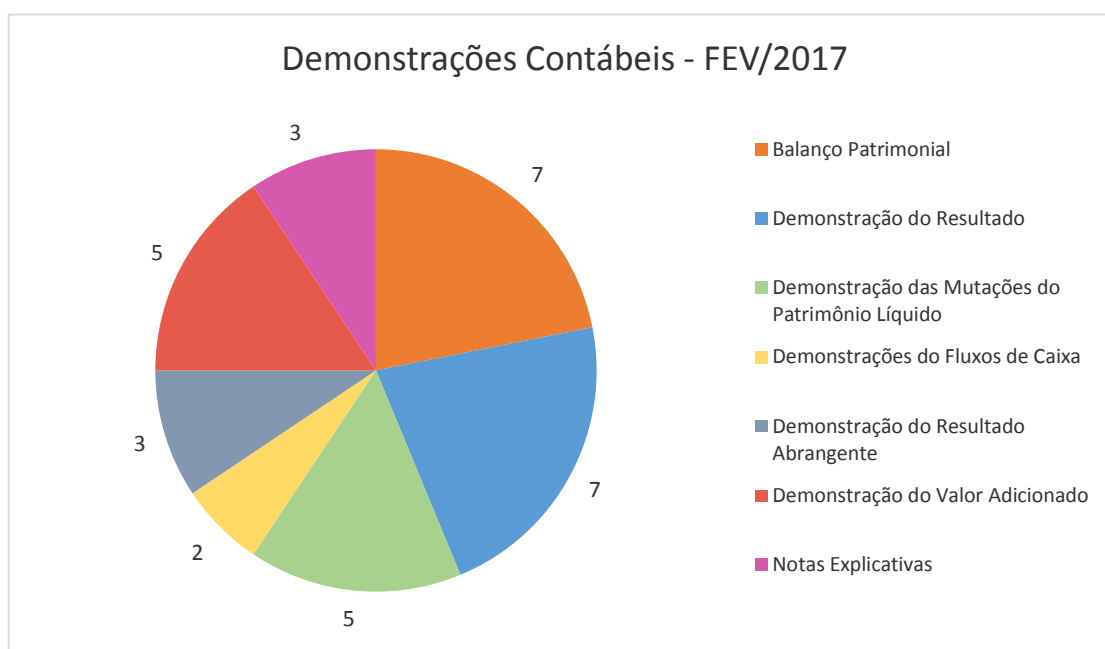


Gráfico 1 – Demonstrações Contábeis
Fonte: RABELO; CANDIDO, 2017

Ao analisar o gráfico e a disponibilidade de funcionários, percebe-se uma previsão de 16 dias utilizados, para cada funcionário, para o fechamento das demonstrações contábeis do cliente em questão, restando somente 04 dias úteis para outros clientes. Onerando significativamente a empresa em função da quantidade de funcionários necessários.

A partir dos problemas custo financeiro e tempo, surge o Sistema de Demonstrações Contábeis, SDC que tem por objetivo ofertar serviços de baixo custo financeiro, rápido e personalizado. Sua principal função será emitir, a partir do plano de contas e balancete da empresa importados os relatórios de demonstrações contábeis, que atendem aos critérios contábeis. O sistema visa proporcionar aos

seus clientes maior foco na tomada de decisões oferecendo relevantes dados gerados sem a necessidade de uma equipe de contabilidade.

O sistema deve ler um arquivo, copiar os dados e armazená-lo para serem manipulados e utilizados na geração de outros relatórios.

3.1.1 VISÃO DO PRODUTO

Para entidades empresariais que tem como necessidade emitir relatórios contábeis, o SDC é uma plataforma web que tem como principal objetivo emitir relatórios de demonstrações contábeis a partir do plano de contas e balancete de forma rápida, barata e precisa, ao contrário das empresas de contabilidade que levam cerca de 20 (vinte) dias para disponibilização dos relatórios contábeis.

3.1.2 ESTÓRIAS DE USUÁRIO

A figura 2 representa o diagrama de caso de uso parcial do SDC:

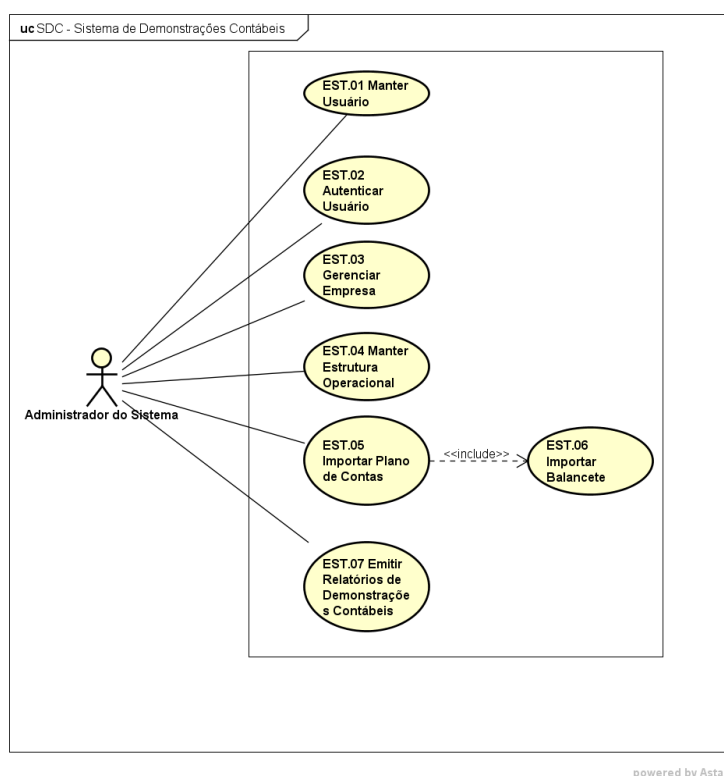


Figura 2 – Diagrama de Caso de Uso SDC
Fonte: RABELO; CANDIDO, 2017

EST01 – Manter Usuário: Tem por finalidade proporcionar ao usuário a funcionalidade de inserir, alterar, ativar, inativar, buscar e visualizar dados de um outro usuário.

EST02 – Autenticar Usuário: Permite aos usuários efetuar *login* no sistema, ou recuperar a sua senha, para acessar as funcionalidades permitidas ao seu perfil.

EST03 – Gerenciar Empresa: Permite ao usuário alterar as informações básicas da entidade empresarial no qual o sistema fará as demonstrações contábeis.

EST04 – Manter Estrutura Operacional: Permite ao usuário inserir, alterar, visualizar e consultar a estrutura operacional do sistema. Podendo configurar as máscaras multiníveis do plano de contas e balancete.

EST05 – Importar Plano de Contas: Realizar a leitura de um arquivo *.csv que contenha o plano de contas da entidade e alimentar o banco de dados do sistema com as informações do arquivo.

EST06 – Importar Balancete: Esta estória permite com que o sistema leia um arquivo *.csv que contenha os dados do balancete da entidade, afim de alimentar o banco de dados com as informações do arquivo.

EST07 – Emitir Relatórios Contábeis: essa estória permite ao usuário, a partir da importação do plano de contas e do balancete, gerar os seguintes relatórios contábeis: Balanço Patrimonial, a Demonstrações dos Fluxos de Caixa, a Demonstração das Mutações do Patrimônio Líquido. Futuramente irá gerar também os seguintes relatórios: a Demonstração do Resultado, a Demonstração do Resultado Abrangente, a Demonstração do Valor Adicionado e as Notas Explicativas.

As descrições das estórias de usuário, assim como os *prints* das telas do sistema desenvolvido se encontram nos anexos C e D respectivamente.

3.2 PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

As metodologias ágeis para desenvolvimento de *software* são uma resposta às chamadas metodologias tradicionais. Mesmo com a evolução dos computadores, das técnicas e ferramentas nos últimos anos, a produção de *software* confiável, correto e entregue dentro dos prazos e custos estipulados ainda é muito difícil, (SANTOS; NETO; 2009).

Um *software* é um produto desenvolvido para atender uma necessidade de negócio, e os seus requisitos mudam a todo instante, logo um *software* constantemente deve ser atualizado para compreender esses novos requisitos, e com isso faz-se necessário que possua uma arquitetura bem definida para que consiga receber essas atualizações

3.3 SCRUM

As equipes possuem uma cultura auto organizável e funcional, escolhem a melhor forma de realizar seu trabalho; ao contrário de serem dirigidas por outros de fora da equipe; têm todas as competências necessárias para realizar o trabalho sem a dependência de outras que não fazem parte da equipe e é projetada para aperfeiçoar a flexibilidade, criatividade e produtividade (SCHWABER; SUTHERLAND, 2011).

O ciclo de vida de um projeto utilizado *Scrum* (exemplificado na figura 3) é focado em identificar impedimentos, por isso ele é focado em padrões preestabelecidos e que funcionaram corretamente no projeto, onde é indispensável que a equipe possua uma grande visão sobre o *software* que está desenvolvendo e saiba introduzir os conhecimentos aprendidos a todos da equipe.

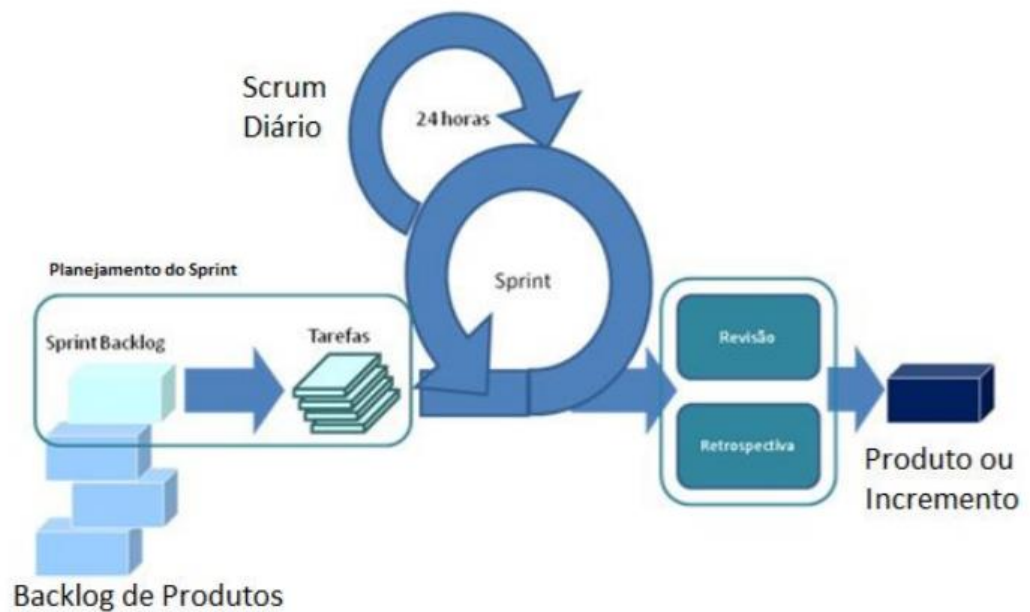


Figura 3 – Ciclo de vida SCRUM
Fonte: (RODRIGUES, 2011)

“[...] *Scrum* aplica-se a projetos tanto pequenos como grandes, esforçando-se para liberar o processo de quaisquer barreiras, o seu principal objetivo é conseguir uma avaliação correta do ambiente em evolução, adaptando-se constantemente ao “caos” de interesses e necessidades, indicado e utilizado para o desenvolvimento de *softwares*” (TOMASEL, 2014, p. 11, *et al*).

O *Framework SCRUM* apenas estabelece conjuntos de regras e práticas de gestão que devem ser adotadas para garantir o sucesso de um projeto. Segundo Ferreira (2005), as principais características do *SCRUM* são:

- É um processo ágil para gerenciar e controlar o desenvolvimento de projetos;
- É um *wrapper* para outras práticas de engenharia de *software*;
- É um processo que controla o caos resultante de necessidades e interesses conflitantes;
- É uma forma de aumentar a comunicação e maximizar a cooperação;
- É uma forma de detectar e remover qualquer impedimento que atrapalhe o desenvolvimento de um produto;
- É escalável desde projetos pequenos até grandes projetos em toda empresa.

A utilização do *scrum* deve-se pelo seu enfoque diferenciado, e a importância dispensada para a documentação em detrimento da programação, onde busca

destrinchar o projeto em ciclos de desenvolvimento, onde no final de seu ciclo terá um produto pronto para utilização do cliente.

3.4 OPENUP

Openup é uma metodologia que segue o processo unificado de engenharia de *software*, porém com uma cultura aberta e enxuta, ele utiliza o movimento ágil a qual foca na natureza colaborativa de desenvolvimento de *software*. A metodologia é formada através de processos simples os quais podem ser expandidos para quaisquer tipos de *softwares*, por conta disso, essa metodologia foi selecionada para fazer parte do processo de desenvolvimento do sistema que será desenvolvido pelos autores.

O *Openup* foi desenvolvido como um projeto de exemplo aplicando as melhores práticas de engenharia de *software*, assim sendo, o *Openup* é um conjunto de metodologias ágeis como *Scrum* e *XP (Extreme Programming)* somado a um pouco da arquitetura do processo unificado.

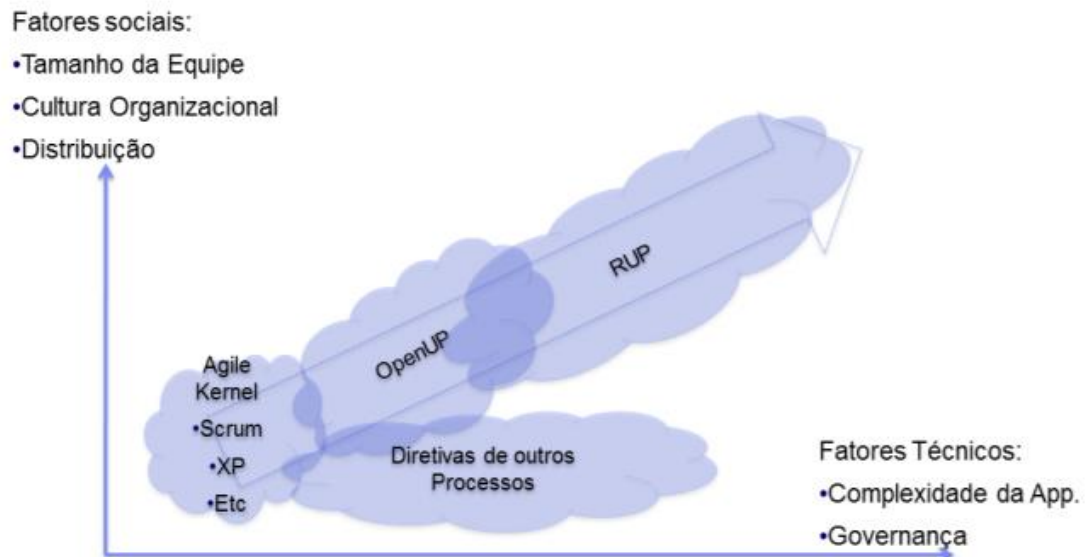


Figura 4 – Perspectiva do Openup
Fonte: (ESSENTIALS, 2011)

Openup possui seu núcleo em desenvolvimento aberto pela comunidade, assim ele pode ser facilmente adaptado para qualquer tipo de desenvolvimento de *software*, torna-se possível também adicionar práticas e

processos únicos aplicados pela cultura organizacional da empresa (DIEDRICH, 2011, P.29).

Através da metodologia é estabelecido um conjunto de papéis, tarefas e artefatos que são utilizados pelos membros da equipe durante a execução e gerenciamento do projeto, conforme figura 5, já conceituados no referencial teórico deste trabalho.



Figura 5 – Disposição do *Openup*
Fonte: (GUSTAFSSON, 2008)

3.4.1 PROCESSO DE DESENVOLVIMENTO HÍBRIDO – SCRUM + OPENUP

A partir das definições e boas práticas empregadas no SCRUM e OPENUP, foi definido a metodologia de desenvolvimento híbrida demonstrada na figura 6 para o desenvolvimento do *software* descrito no capítulo anterior.

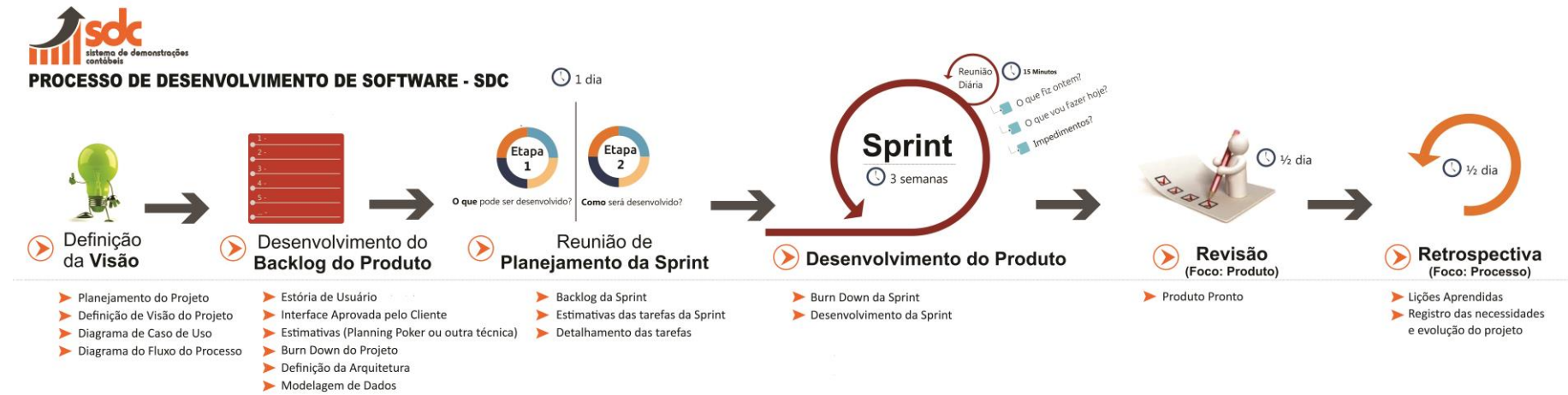


Figura 6 – Processo de Desenvolvimento de Software
Fonte: RABELO; CANDIDO; 2017.

O Processo de desenvolvimento foi dividido em 6 (seis) partes, são eles:

- **Definição da Visão:** Nesta fase, os membros da equipe discutem e planeja o projeto. É definido a visão do produto, os seus principais requisitos e seu processo. Os artefatos gerados nesta etapa são o diagrama de caso de uso parcial e o diagrama de fluxo de processo.
- **Desenvolvimento do *Backlog* do Produto:** O *Product Owner* (P.O.) elabora as histórias de usuário e interfaces funcionais a partir das informações do cliente. Em seguida, é validada a corretude das informações e listada as histórias no *Backlog* do Produto. Juntamente com o time, o P.O. define a arquitetura do sistema e a modelagem de dados.
- **Reunião de Planejamento da *Sprint*:** O P.O. juntamente com o time de desenvolvimento, subdivide os itens do *backlog* do produto em atividades que compõem a *sprint*. Em seguida, o time de desenvolvimento estima o tempo gasto para cada atividade.
- **Desenvolvimento do Produto:** No decorrer de 3 semanas as atividades do *backlog* da *sprint* são realizadas, nesta fase do processo. Diariamente é realizada uma reunião denominada reunião diária, onde os membros do time informam como está o andamento de suas atividades.
- **Revisão:** Nesta fase do processo, o time de desenvolvimento apresenta o que foi desenvolvido no decorrer da *sprint* para o P.O. e o cliente, afim de verificar se o que foi desenvolvido está conforme o especificado nas histórias de usuário.
- **Retrospectiva:** Ocorre após a revisão, onde afim de melhorar o processo e o ambiente de trabalho, o time de desenvolvimento apresenta ao *scrum master* o que foi bom durante a *sprint* e o que precisa melhorar.

3.5 ARQUITETURA DO SOFTWARE

A figura 7, representa a arquitetura do *software* que será desenvolvido:

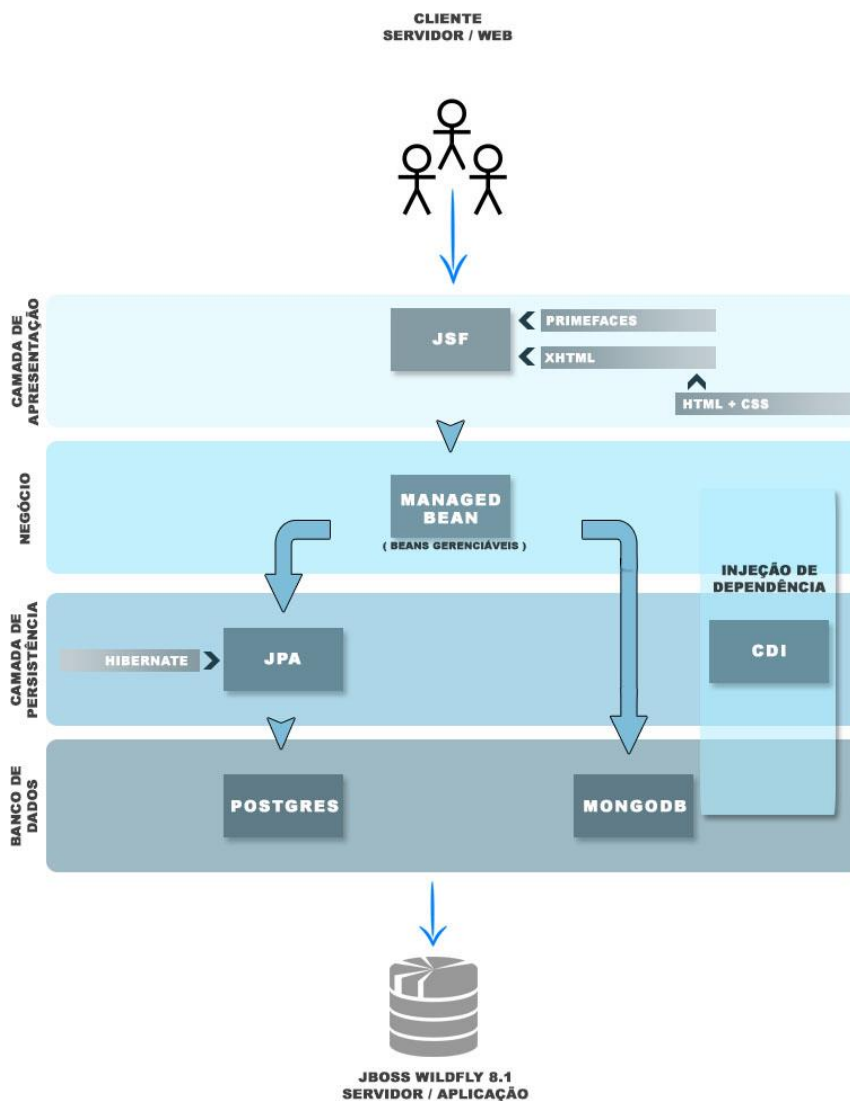


Figura 7 – Arquitetura do *software*
Fonte: RABELO,; CANDIDO, 2017.

Na arquitetura apresentada acima, o cliente irá realizar uma requisição através da camada de apresentação (JSF, somado ao *Primefaces* e *XHTML*), onde na camada de negócio, através do *Managed Bean*, que é responsável por dispor os dados e métodos que possam ser referenciados pela camada de apresentação para

entrada de dados, apresentação dos dados ou tratamento de eventos, enviará através da camada de persistências os dados para os bancos de dados.

O SGBD relacional definido foi o *PostgreSQL*. Que será utilizado para controle de acesso de usuário no sistema, propiciando um maior grau de vigor quanto à consistência de informações e ainda no armazenamento dos relatórios importados, para as verificações em relação ao desempenho.

Outros SGBDR's poderiam ser utilizados, porém o *SQLServer*, *DB2* e o *Oracle* possuem um alto custo para sua aquisição, apesar de suas muitas vantagens, o preço da licença de uso inviabiliza a utilização do mesmo para o projeto.

Em relação aos SGBD's não relacionais, dentre suas classificações, será utilizado para o desenvolvimento o SGBD orientado à documentos. O *software* irá armazenar e recuperar os relatórios importados em forma de documento. Entre os diversos SGBD's não relacionais destacam-se o *MongoDB*, *CouchDB* e *RavenDB*.

O *MongoDB* foi selecionado para compor a arquitetura do sistema pois armazena seus dados sem transações e sem *joins*, resultando em consultas mais simples, além disso possui licença gratuita. Possui alta *performance*, sem tabela (*schema*) e construído em escala horizontal, armazena todas as informações em um documento (*XML*).

O *CouchDB* apesar de sua grande escalabilidade e flexibilidade, é desestruturado trazendo uma falta de esquema rígido.

O *RavenDB* apesar de ter a versão gratuita para uso não comercial caso faça parte de um *software* proprietário, é necessário adquirir a licença, e por este motivo, torna-se inviável.

3.6 BENEFÍCIOS E MALEFÍCIOS DO BANCO DE DADOS NÃO RELACIONAL

Segundo Pauleto, Pains, Dantas e Sakurai (2013) “os bancos de dados relacionais dominaram o mercado de armazenamento de informações por diversos anos”, porém com a necessidade de armazenar grandes volumes de dados, surge um atual interesse no uso dos bancos de dados não-relacionais para armazenamento de dados.

A principal característica dos bancos de dados não-relacionais, é o afastamento do modelo de banco de dados relacional. Enquanto no banco de dados relacional os dados são armazenados em tabelas e ligados entre si por relacionamentos, no banco de dados não-relacional “cada solução *NoSQL* possui um modelo diferente” (SADALAGE, FOWELER, 2013, p.13). Este modelo de banco de dados permite uma maior escalabilidade, performance e disponibilidade na manipulação dos dados.

As características e recursos propiciou ao modelo relacional de banco de dados uma posição de destaque e predominância nos diversos ambientes computacionais. No entanto, a sua complexidade estrutural fez com que surgissem problemas, principalmente relacionados ao crescente volume de dados que as empresas necessitam armazenar atualmente (DEV MEDIA, 2016).

“De um modo geral, os principais problemas encontrados com a utilização do modelo relacional estavam concentrados na dificuldade em se conciliar tal modelo com a demanda por escalabilidade cada vez mais frequente” (BRITO, 2010).

Um sistema que armazena dados em um banco de dados relacional é natural uma queda de performance, pois quanto mais registros no banco relacional maior o tempo necessário para processar os dados. “Para superar esse problema, restam duas alternativas, promover um *upgrade* no servidor ou aumentar o número de servidores” (BRITO, 2010).

Os SGBD's relacionais não se adaptam muito bem a escalabilidade horizontal, e apesar de aplicar a escalabilidade vertical pode não ser suficiente caso o aumento de registros continue sendo gradativo, pois o problema passa a ser no acesso à base de dados.

Quando se opta por uma estratégia *NoSQL* para substituir a utilização de um SGBD tradicional, deve-se levar em conta algumas observações como falta das propriedades ACID, modelo de dados, desempenho e ferramentas.

Estes novos bancos de dados trocam algumas propriedades Atomicidade, Consistência, Isolamento e Durabilidade (ACID), propriedades que sustentam os bancos de dados relacionais, por um modelo de dados com propriedade Basicamente Disponível, Estado Leve e Eventualmente Consistente (BASE).

Os SGBD's não relacionais “premia a disponibilidade em detrimento a consistência, com o foco em arquiteturas de dados altamente escaláveis e acessíveis” (MERIAT, 2013).

Outro aliado ao banco de dados não relacional, é a falta de esquema, que garante escalabilidade e aumento da disponibilidade, proporcionando ao mesmo uma maior eficiência em suas consultas. A falta de esquema remete a não verificação de integridade e de relacionamentos.

Para aumentar a eficiência do banco de dados não-relacionais, basta investir em clusters, escalabilidade horizontal, quanto mais clusters maior será o desempenho no banco de dados. Outra vantagem na utilização de clusters é a disponibilidade de ferramentas que gerenciam clusters e escalabilidade.

3.7 MÉTRICAS DE DESEMPENHO DE BANCO DE DADOS NÃO-RELACIONAL

A crescente necessidade de armazenar mais e mais dados verifica-se a necessidade de utilização de SGBDs melhores e mais eficientes. Segundo Carneiro, Moreira e Freitas (2010), “realizar operações, de forma eficiente, sobre estas grandes coleções de dados é uma questão fundamental, já que o desempenho de um SGBD é medido a partir de sua eficiência diante de consultas e alterações”.

“Não existe um meio universal com o qual possamos avaliar o desempenho dos diversos sistemas computacionais, é necessário definir técnicas e métricas para

cada caso” (FERREIRA, JÚNIOR, 2012, apud GUIMARÃES, 2002). Este conceito se aplica também à banco de dados, pois há vários SGBD’s diferentes disponíveis no mercado e cabe ao desenvolvedor analisar e selecionar o SGBD ideal.

No decorrer deste trabalho, foi analisado um *software* que armazena seus dados em um banco de dados relacional (*PostgreSQL*) e um não relacional (*MongoDB*). Estes dois SGBD’s serão comparados de forma objetiva a partir das métricas de desempenho definidas neste capítulo.

3.7.1 DEFINIÇÕES DAS MÉTRICAS DE DESEMPENHO

Segundo Microsoft TechNet, “a avaliação contínua do desempenho de banco de dados ajuda a minimizar tempos de resposta e a maximizar a taxa de transferência, permitindo alcançar desempenho ótimo”. Para definir as métricas de desempenho será levado em conta três (3) pilares: Velocidade, Consistência e Disponibilidade.

3.7.1.1 Métricas de Velocidade

Segundo Leal (2016), “se o sistema executa o serviço corretamente, seu desempenho é medido pelo tempo necessário para executar o serviço, a taxa na qual o serviço é realizado e os recursos consumidos durante a execução do serviço”. Essas três medidas podem ser melhor definidas como: tempo de resposta, vazão e utilização, respectivamente.

3.7.1.1.1 *Tempo de Resposta*

Segundo Nova Solução (2017), “a análise do tempo de resposta é uma abordagem eficaz para melhorar o desempenho do banco de dados”. O tempo de resposta é definido como o intervalo entre o envio de uma requisição ao banco de dados e a resposta do banco de dados ao usuário, conforme figura abaixo.



Figura 8 – Tempo de resposta
Fonte: LEAL, 2016

O tempo de resposta é fortemente influenciado pela quantidade de dados solicitados ou enviados pelo usuário. Leal (2016) afirma que, “o tempo de resposta de um sistema geralmente aumenta à medida que a carga do sistema aumenta”.

3.7.1.1.2 Vazão ou *Throughput*

Vazão ou *Throughput*, segundo Ikematu:

“Define a capacidade do computador de processar os dados. Ele é uma composição de velocidade de I/O, velocidade da CPU, capacidades de paralelismo da máquina e a eficiência do sistema operacional e o *software* básico envolvido”.

Ou seja, utilização de recursos com baixa velocidade do processador, espaço em disco baixo, sistema operacional, cache e memória afetam no desempenho do banco de dados.

3.7.1.1.3 Utilização

“A utilização (*utilization*) de um recurso é medida como a fração do tempo utilizado pelo recurso em relação às demais solicitações de serviço. Por exemplo, a razão entre o tempo ocupado e tempo total durante um determinado período. O período durante o qual o recurso não está sendo utilizado é chamado de tempo ocioso (*idle time*)” (LEAL, 2016).

Deve-se projetar banco de dados pensando em equilibrar a utilização dos recursos. Segundo Afranio, “a utilização dos recursos, bem como o processamento

interno do SGBD pode ser monitorada para revelar gargalos, tais como a disputa pelos mesmos dados por vários usuários ou processos simultaneamente”.

3.7.1.2 Consistência

Para que um sistema de banco de dados sempre possa fornecer informações confiáveis aos usuários, é necessário que o banco de dados mantenha uma consistência em seus dados a fim de evitar a ocorrência de possíveis erros relacionados aos mesmos.

3.7.1.3 Disponibilidade

Segundo a norma ABNT ISO/IEC 9126-1 (2003) disponibilidade é “a capacidade de um produto de *software* de estar pronto para executar uma função requisitada num dado momento, sob condições especificadas de uso”. De forma resumida, a disponibilidade é definida pelo tempo que o sistema e o banco de dados estarão disponível para atender as requisições dos usuários.

3.8 CHECKLIST DE ANÁLISE DE DESEMPENHO DO BANCO DE DADOS

Baseado nas métricas apresentadas no subitem anterior, foi desenvolvido o *checklist* abaixo, figura 9, para analisar o desempenho do banco de dados:

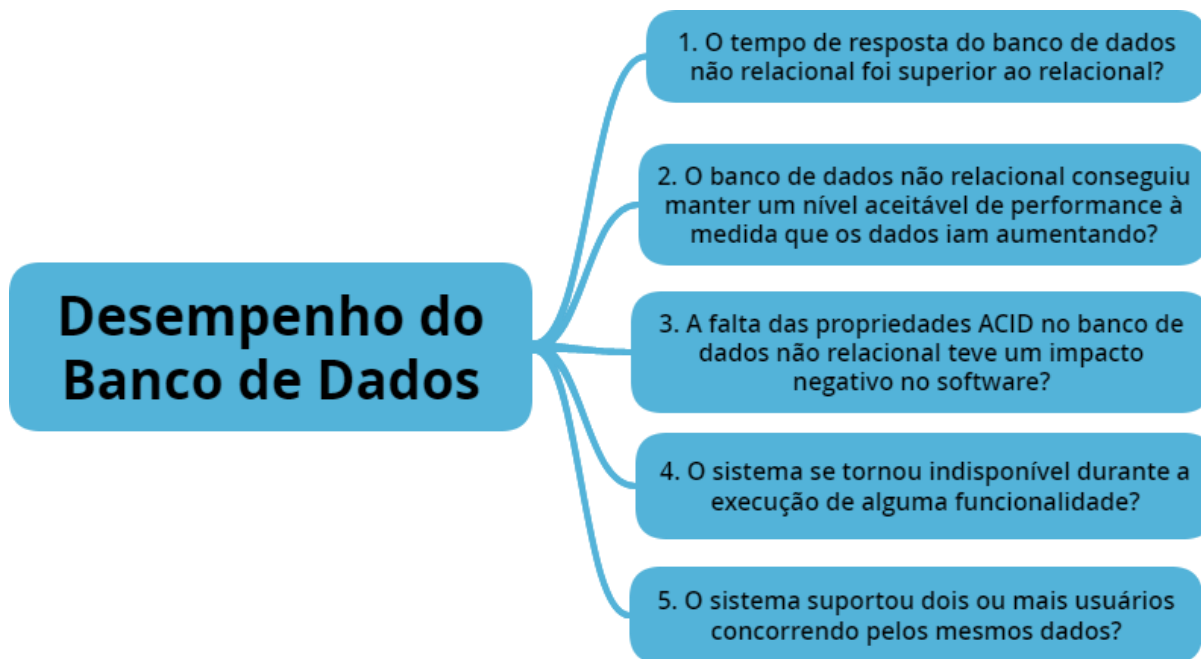


Figura 9 – *Checklist* de Desempenho do Banco de Dados
Fonte: RABELO; CANDIDO, 2017.

3.9 DESENVOLVIMENTO DA APLICAÇÃO

Neste item será discutido o processo de implementação do *software* SDC, utilizado para realização dos testes de desempenho do banco de dados não relacional.

A aplicação foi desenvolvida no ambiente de desenvolvimento *eclipse neon* 4.6.3, com servidor *wildfly* versão 8.1, utilizando um kit de desenvolvimento *Java* na versão 1.8.0 (conhecido como *JDK*). Além disso, a aplicação utiliza conceitos de orientação a objetos em sua estrutura. Dos 4 (quatro) pilares da orientação à objetos: abstração, encapsulamento e herança e polimorfismo. Abaixo são apresentados algumas classes e métodos do sistema onde se verifica alguns dos pilares:

```
28
29- /**
30  * Controlador para o Caso de Uso - De autenticação de usuário.
31  *
32  * @author Macro.
33  */
34 @ManagedBean
35 @ViewScoped
36 > public class AutenticarUsuarioController extends AbstractController {
37
38     private static final long serialVersionUID = -2348133914536828976L;
39
40- @ManagedProperty(value = "#{authenticationManager}")
41     private AuthenticationManager authenticationManager;
42
43- @Inject
44     private LoginService loginService;
45
46     private Usuario usuario;
47
48     private AcaoLogin acaoLogin;
49
50     private String email;
51     private String senha;
52     private String novaSenha;
53     private String confirmacaoSenha;
54
```




Figura 10 – Herança
Fonte: RABELO; CANDIDO, 2017.


```

26
27 /**
28  * Entidade que representa a tabela <b>MASCARA</b>.
29  *
30  * @author Macro.
31  */
32 @Entity
33 @Table(name = "MASCARA")
34 public class Mascara implements Entidade {
35
36     private static final long serialVersionUID = -3574483495280650512L;
37
38     @Id
39     @Column(name = "ID_MASCARA")
40     @GeneratedValue(strategy = GenerationType.IDENTITY)
41     private Long id;
42
43     @Column(name = "MASCARA")
44     private String mascara;
45
46     @Column(name = "NOME")
47     private String nome;
48
49     @OneToMany(fetch = FetchType.LAZY, mappedBy = "mascara", cascade = CascadeType.ALL, orphanRemoval=true)
50     private List<Nivel> niveis;
51
52     @Transient
53     private boolean excluido;
54
55     /**
56     * @return the id
57     */
58     public Long getId() {
59         return id;
60     }

```

ABSTRAÇÃO

ENCAPSULAMENTO

Figura 11 – Abstração e encapsulamento
Fonte: RABELO; CANDIDO, 2017.

Além do kit do JDK, foram utilizadas tecnologias como *Java Server Faces* (JSF) e *Primefaces* para o desenvolvimento das interfaces da aplicação. O JSF é uma especificação *Java* para construção de interfaces baseadas em componentes para web. E o *primefaces* é um componente do JSF que facilita na implementação da interface.

Em relação aos bancos de dados foram selecionados, conforme já mencionado anteriormente, as tecnologias *PostgreSQL* e *MongoDB*. Para os autores a tecnologia *MongoDB* era inédita, naturalmente, obtiveram uma maior dificuldade para utilização da mesma, pois tiveram que adquirir o conhecimento técnico referente à tecnologia para poder utilizá-la.

A maior dificuldade encontrada na utilização do *MongoDB* foi na instalação e configuração do banco de dados, já a utilização de seus comandos, como por exemplo inserção, alteração, exclusão dentre outros, foi de fácil entendimento. Levando em consideração ainda que o modelo não relacional e seus comandos eram desconhecidos para os autores.

3.10 EXECUÇÃO DOS TESTES

A partir do *checklist* elaborado, realizou-se testes de carga em ambos os bancos de dados - com os mesmos arquivos de testes - através da utilização da aplicação mencionada nos capítulos anteriores. Os resultados destes testes serão demonstrados a seguir.

3.10.1 AMBIENTE DE SIMULAÇÃO

Os testes executados tiveram como ambiente de execução um *notebook Dell Inspiron 15R* com processador core i7 2,9GHz com 8GB de memória RAM, 1TB de HD, 8GB de SSD e placa de vídeo dedicada GTXforce 2GB, com sistema operacional Windows 10 pro. Todos os testes realizados foram efetuados em banco de dados local.

3.10.2 TESTE DE CARGA

Afim de averiguar o tempo de resposta de cada um dos bancos de dados, foi executado um teste de carga nas funcionalidades de importar/visualizar plano de contas, importar balancete e geração de relatórios (balanço patrimonial, demonstração de fluxos de caixa e demonstrações das mutações do patrimônio líquido). Foram feitas três rodadas de testes, em ambos os bancos de dados com os mesmos arquivos, onde o primeiro leu um arquivo de 1674 registros, o segundo 10.000 registros e o último 100.000 registros. O processo para execução dos testes foi feito nas seguintes etapas:

- Ao acessar o sistema e executar a funcionalidade, foi aplicado um comando de inserção ou seleção no banco de dados (depende da funcionalidade executada), obtendo, assim, o primeiro resultado. Os

resultados foram obtidos através do console do ambiente de desenvolvimento.

- O computador é reiniciado, para garantir que não há interferência na memória, processamento ou até mesmo cash de dados.
- Ao final de todos os testes com o primeiro banco de dados (1674 registros), ambos os bancos (relacional e não relacional) foram deletados e os dados referente aos 10000 registros foram inseridos.
- Repete-se todo o processo, agora com 10000 registros. Logo em seguida foi feito com 100000 registros.

Os resultados dos testes podem ser visualizados nas tabelas a seguir (evidências que comprovam os testes realizados se encontrar no anexo B):

Tabela 1 – Importar Plano de Contas

	1674 registros	10.000 registros	100.000 registros
Banco de Dados Não Relacional (MongoDB)	156 ms	878 ms	7112 ms
Banco de Dados Relacional (Postgressql)	1375 ms	61707 ms	14770 ms

Fonte: RABELO; CANDIDO, 2017.

Tabela 2 – Visualizar Plano de Contas

	1674 registros	10.000 registros	100.000 registros
Banco de Dados Não Relacional (MongoDB)	2 ms	2 ms	3 ms
Banco de Dados Relacional (Postgressql)	7 ms	12 ms	43 ms

Fonte: RABELO; CANDIDO, 2017.

Tabela 3 – Importar Balancete

	1674 registros	10.000 registros	100.000 registros
Banco de Dados Não Relacional (MongoDB)	54 ms	487 ms	4787 ms
Banco de Dados Relacional (Postgressql)	147 ms	1427 ms	10692 ms

Fonte: RABELO; CANDIDO, 2017.

Tabela 4 – Gerar relatório de balanço patrimonial

	1674 registros	10.000 registros	100.000 registros
Banco de Dados Não Relacional (MongoDB)	1 ms	1 ms	2 ms
Banco de Dados Relacional (Postgressql)	2 ms	4 ms	23 ms

Fonte: RABELO; CANDIDO, 2017.

Tabela 5 – Gerar relatório de demonstração de fluxo de caixa

	1674 registros	10.000 registros	100.000 registros
Banco de Dados Não Relacional (MongoDB)	1 ms	1 ms	5 ms
Banco de Dados Relacional (Postgressql)	4 ms	5 ms	23 ms

Fonte: RABELO; CANDIDO, 2017.

Tabela 6 – Gerar relatório de demonstração das mutações do patrimônio líquido

	1674 registros	10.000 registros	100.000 registros
Banco de Dados Não Relacional (MongoDB)	1 ms	2 ms	3 ms
Banco de Dados Relacional (Postgressql)	6 ms	19 ms	36 ms

Fonte: RABELO; CANDIDO, 2017.

Em todos os cenários, constatou-se que o banco de dados não relacional obteve um desempenho superior em relação ao relacional. Percebeu-se que na medida que o número de linhas dos arquivos importados aumentava, o banco de dados relacional apresentava uma queda de desempenho. Já o banco de dados não relacional, apesar de ter uma queda de desempenho, o sistema executou suas funcionalidades em média de 0,897 segundos (tempo calculado através de uma média aritmética dos valores obtidos), o que leva a concluir-se que a medida que o

banco de dados do sistema for “inflando” o banco de dados não relacional terá um desempenho melhor.

4 RESULTADOS ALCANÇADOS

A finalidade deste trabalho foi analisar o desempenho de banco de dados em um ambiente, que inicialmente, haverá armazenamento de poucos dados, onde os dados foram armazenados em um banco de dados relacional e não relacional.

Para isto, um dos objetivos foi desenvolver uma aplicação que simula este ambiente, foi desenvolvido então o SDC. Para implementação da aplicação foi analisado os tipos de bancos de dados *NoSQL* disponíveis para utilização. Foram analisados os tipos de Chave-valor, orientado à documentos, família de colunas e grafos.

A análise desenvolvida sobre os tipos de *NoSQL* teve a intenção de adquirir conhecimento acerca de como estes bancos de dados funcionam e qual deveria ser selecionado para compor a arquitetura do sistema, para que após a implementação da aplicação o banco seja analisado conforme as métricas propostas no decorrer deste trabalho. Após a realização da análise dos bancos de dados, o *MongoDB* foi selecionado levando em consideração as informações obtidas.

Os bancos de dados relacionais também foram analisados, e o *PostgreSQL* foi selecionado devido sua licença e os recursos disponíveis para este projeto.

A aplicação proposta foi desenvolvida, após levantamento e especificação de requisitos, na linguagem de programação *Java* e o principal objetivo dela é importar arquivos de plano de contas e balancete e emitir os principais relatórios de demonstrações contábeis. A aplicação armazena os dados de plano de contas e balancete na base de dados do *MongoDB* e *PostgreSQL*, já o controle de acesso e cadastro de usuários apenas no *PostgreSQL*.

Outro objetivo proposto era realizar uma comparação dos benefícios e malefícios dos bancos de dados não relacionais e traçar métricas de desempenho dos bancos de dados não relacionais. Estes estudos tinham como finalidade levantar pontos para compor um *checklist* de avaliação de desempenho de um banco de dados *NoSQL*, para que posteriormente este mesmo *checklist* fosse utilizado na

aplicação desenvolvida pelos autores para que o banco de dados pudesse ser analisado.

Neste ponto, onde os autores tinham em mãos a aplicação e o *checklist*, foram realizados testes de carga nos quais obtiveram o seguinte resultado do *checklist*:

Tabela 7 – Resultado do Checklist de Análise de Desempenho

Pergunta	Conclusão
O tempo de resposta do banco de dados não relacional foi superior ao relacional?	Sim, em todos os cenários de testes propostos o banco de dados não relacional obteve um desempenho superior ao relacional.
O banco de dados não relacional conseguiu manter um nível aceitável de performance à medida que os dados iam aumentando?	Sim. Foi realizado testes com três quantidades de dados no banco: 1674, 10000 e 100000, em todos os casos, o banco de dados obteve um tempo de resposta aceitável, não passando de 14 segundos.
A falta das propriedades ACID no banco de dados não relacional teve um impacto negativo no <i>software</i> ?	Sim, a falta de propriedades ACID como por exemplo a atomicidade faz com que não garanta a integridade dos dados no banco de dados. Na aplicação, a importação do plano de contas e balancete é de vital importância para manter a confiabilidade dos relatórios emitidos. No teste onde a comunicação com o banco de dados foi perdida na importação do plano de contas fez com que apenas 5 registros do plano de contas fosse armazenado no banco de dados, logo se o usuário não percebesse que os dados estavam incompletos, os relatórios emitidos pelo sistema estariam incorretos. Porém, este problema foi facilmente tratado através de uma validação que apresentará uma mensagem ao usuário quando este problema acontecer.
O sistema se tornou indisponível durante a execução de alguma funcionalidade?	Não, o banco de dados permaneceu acessível a outros usuários durante a execução das funcionalidades dos mesmos.
O sistema suportou dois ou mais usuários concorrendo pelos mesmos dados?	Sim, foi realizado teste onde dois usuários de forma simultânea solicitaram visualizar um plano de contas. O sistema conseguiu dispor dados para ambos os usuários.

Fonte: RABELO; CANDIDO, 2017.

Com a execução deste trabalho chegou-se na conclusão de que a utilização de banco de dados não relacionais para sistema que não sejam *big data* se torna viável.

No entanto, o cenário descrito neste trabalho, apesar do banco de dados não relacional ter apresentado um tempo de resposta superior ao banco de dados relacional, a falta das propriedades ACID, propriedades não encontradas no *NoSQL*, deixou o *software* sem confiabilidade, porém, esta situação foi facilmente contornado através de uma validação que apresenta ao usuário uma mensagem quando houve erro nas importações dos arquivos com isso, basta o usuário importar novamente os arquivos.

Porém, por ser um *software* que, após anos de utilização, armazenará uma quantidade elevada de dados e a necessidade de geração de diversos relatórios com tempo de resposta quase que imediata, o uso do modelo não relacional - *NoSQL* - é mais aconselhável do que o relacional.

Como sugestão para trabalhos futuros, indica-se a implementação dos demais relatórios de demonstrações contábeis.

REFERÊNCIAS

ABCARTICULOS. **Vantagens e Desvantagens do MySQL**. Disponível em: < <http://pt.abcarticulos.info/article/vantagens-e-desvantagens-do-mysql>>. Acesso em: 27/06/2017.

ABNT ISO/IEC 9126-1. **Engenharia de Software – Qualidade de produto**. 2003. Disponível em: < http://jkolb.com.br/wp-content/uploads/2014/02/NBR-ISO_IEC-9126-1.pdf>. Acesso em: 15/11/2017

AFRANIO, R. **Performance de banco de dados MySQL**. Disponível em: < <https://www.devmedia.com.br/performance-do-banco-de-dados-mysql/18508>>. Acesso em: 15/11/2017.

ARAÚJO, E. C. **Desenvolvimento para WEB com Java**. Florianópolis: Visual Books Editora, 2010.

BRITO, R. W. **Banco de Dados NoSQL x SGBDs Relacionais: Análise Comparativa**. Fortaleza: Infobrasil, 2010. Disponível em: < <http://www.infobrasil.inf.br/userfiles/27-05-S4-1-68840-Bancos%20de%20Dados%20NoSQL.pdf> >. Acesso em: 23 maio 2017.

CARNEIRO, A. P.; MOREIRA, J. L.; FREITAS, A. L. C. **TUNING – Técnicas de otimização de Banco de Dados um estudo comparativo: MySQL e PostgreSQL**. Rio Grande, 2010. Disponível em: <<http://repositorio.furg.br/bitstream/handle/1/1692/TUNING.pdf?sequence=1>>. Acesso em: 15/11/2017.

COCKBURN, A.; HIGHSMITH, J. **Agile software development: The business of innovation**. Los Angeles: Boehm, 2001.

COELHO, O. P. **Arquitetura: Princípios para alcançar desempenho e escalabilidade em Aplicações**. 2004. Disponível em : <https://msdn.microsoft.com/pt-br/library/cc518051.aspx>. Acessado em: 20/11/2017

DATE, C. J. **Introdução a sistemas de bancos de dados**. 8. ed. Rio de Janeiro: Elsevier, 2003.

DEVMEDIA. **Comparando o NoSQL ao modelo relacional**. 2016. Disponível em: <<https://www.devmedia.com.br/comparando-o-NoSQL-ao-modelo-relacional/30917>> Acesso em: 15/11/2017.

DIEDRICH, L.G. **Integração da Metodologia Ágil *Openup* nos processos de engenharia de software**. Medianeira, 2011. Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/1118/3/MD_ENGESS_I_2012_14.pdf>. Acesso em: 24/05/2017.

ELMASRI; NAVATHE. **Sistemas de banco de dados**. 6. ed. São Paulo: Pearson, 2011.

FERREIRA, D. **SCRUM – Um Modelo Ágil para Gestão de Projetos de Software**. Disponível em: <http://paginas.fe.up.pt/~aaguiar/es/artigos%20finais/es_final_19.pdf>. Acesso em: 15/03/2017.

FERREIRA, E. R.; JÚNIOR, S. M. T. **Análise de desempenho de banco de dados**. Disponível em: <<http://www.unipac.br/site/bb/tcc/tcc-15ee06c022b6b866f2815b76757c667f.pdf>>. Acesso em: 15/11/2017.

FERNANDES, A. S. **NoSQL Base x ACID teorema CAP**. Disponível em: <<https://pt.slideshare.net/Celio12/trabalho-no-sql-aricelio-de-souza>>. Acesso em: 15/03/2017.

FURLAN, D. A. **Banco de Dados: Criação de ambiente Linux, migração de dados e desenvolvimento PL/SQL**. Fundação Educacional do Município de Assis – FEMA – Assis, 2015.

GARLAN, D.; SHAW, M. *An introduction to software architecture*. Pittsburgh: Carnegie Mellon University, 1994.

GARLAN, D., **Software Architecture: a Roadmap**. Limerick, Ireland, 2000.

GIBSON, A. **Trabalhando com Injeção e Qualificadores no CDI**. 2016. Disponível em <https://netbeans.org/kb/docs/javaee/cdi-inject_pt_BR.html>. Acesso em: 20/11/2017.

HEUSER, C. A. **Projeto de Banco de Dados**. 4. ed. Porto Alegre: Sagra, 1998.

IKEMATU, R. S. **Realizando Tuning na Base de Aplicações**. Disponível em: <<http://www.batebyte.pr.gov.br/modules/conteudo/conteudo.php?conteudo=1592>>. Acesso em: 15/11/2017.

JÚNIOR, A. V. R.; JUNIOR, E. A. O. **Análise de desempenho dos frameworks de persistências hibernate e eclipselink**. Maringá, 2014. Disponível em: <<http://www.espweb.uem.br/site/files/tcc/2012/N%C3%A3o%20pagaram/Artigo%20EspWeb%202013%20-%20Aur%C3%A9lio%20Junior.pdf>>. Acesso em: 23 maio 2017.

KITO, D. **Javaserwer faces in action**. Mann Foreword: Burns, 2005.

LEAL, B. G. **Avaliação de Desempenho de Sistemas**. 2016. Disponível em: <<http://www.univasf.edu.br/~brauliro.leal/ensino/ADS/ADS.pdf>>. Acesso em: 15/11/2017.

LUNELLI, R. L. **Demonstrações Contábeis segundo o IFRS**. Portal da Contabilidade. Disponível em: <<http://www.portaldecontabilidade.com.br/tematicas/demonstracoescontabeisifrs.html>>. Acesso em: 23 maio 2017.

MERIAT, V. **Azure Table Storage e o NoSQL – Conceitos, ACID, BASE e o Big Data**. Disponível em: <<http://www.vitormeriat.com.br/2013/12/16/azure-table-storage-e-o-NoSQL-conceitos-acid-base-e-o-big-data/>>. Acesso em: 20/11/2017.

MICROSOFT TECHNET. **Avaliando o desempenho**. Disponível em: <[https://technet.microsoft.com/pt-br/library/ms179440\(v=sql.105\).aspx#Nesta seção](https://technet.microsoft.com/pt-br/library/ms179440(v=sql.105).aspx#Nesta%20se%C3%A7%C3%A3o)>. Acesso em: 15/11/2017.

MORO, A. **Openup**. Disponível em: <http://epf.eclipse.org/wikis/openuppt/openup_basic/customcategories/introduction_t_o_openup_basic,_BTJ_YMXwEduywMSzPTUUwA_1.html>. Acesso em: 23 maio 2017.

MOURA, A. V.; CASANOVA, M. A. **Princípios de sistemas de gerência de bancos de dados distribuídos**. 1999. Disponível em: <<http://www-di.inf.puc-rio.br/~casanova/Publications/Books/1985-BDD.pdf>>. Acesso em: 23 maio 2017.

NOVA SOLUÇÃO. **Análise do tempo de resposta: como melhorar o desempenho do banco de dados medindo a experiência do usuário**. 2017.

Disponível em: <<http://www.anovasolucao.net/analise-do-tempo-de-resposta/>>. Acesso em: 15/11/2017.

SANTOS, V. M.R.; NETO, J. S. C. **Adaptação da Metodologia JAD para o Desenvolvimento de Games.** 2009. Disponível em: <<http://www.eventosufrpe.com.br/jepex2009/cd/resumos/R1152-1.pdf>>. Acesso em: 23 junho 2017.

STEPPAT, N. **Bancos de dados não relacionais e o movimento NoSQL.** 2009. Disponível em: <<http://blog.caelum.com.br/bancos-de-dados-nao-relacionais-e-o-movimento-NoSQL/>>. Acesso em: 24/05/2017.

OLIVEIRA, D. **Analytics: comece pequeno e depois amplie, aconselha IBM.** Itforum365, 2015. Disponível em: <<http://www.itforum365.com.br/analytics/analytics-comece-pequeno-e-depois-amplie-aconselha-IBM/>>. Acesso em: 20 maio 2017.

OLIVEIRA, H. P. **O que é o PostgreSQL ? PostgreSQL .** Disponível em: <<http://pgdoctbr.sourceforge.net/pg82/intro-what-is.html>>. Acesso em: 20 maio 2017.

OLIVEIRA, S. S. **Bancos de dados não-relacionais: Um novo paradigma para armazenamento de dados em sistemas de ensino colaborativo.** Amapá, 2014. Disponível em: <<http://www2.unifap.br/oliveira/files/2016/02/35-124-1-PB.pdf>>. Acesso em: 23/05/2017.

ORACLE. **O banco de dados de código aberto mais conhecido do mundo.** Disponível em: <<https://www.oracle.com/br/mysql/index.html>>. Acesso em: 27/06/2017.

PAULETO, PAINS, DANTAS E SAKURAI. **Estudo comparativo das tecnologias de banco de dados relacional vs. NoSQL (MySQL vs. MongoDB).** São Paulo, 2013. Disponível em <<https://www.metodista.br/congressos-cientificos/index.php/CM2013/facs/paper/view/5227>>. Acesso em: 15/11/2017.

PORCELLI, A. **O que é NoSQL?** Disponível em: <<http://www.devmedia.com.br/o-que-e-NoSQL-java-magazine-86/18777>>. Acesso em: 15/03/2017.

PORTAL DA CONTABILIDADE. **Relatórios de Demonstrações Contábeis.** Disponível em:

<<http://www.portaldecontabilidade.com.br/tematicas/demonstracoescontabeis.htm>>. Acesso em: 23/06/2017.

SADALAGE, P. J.; FOWLER, M. **NoSQL Essencial**: Um guia conciso para o mundo emergente da persistência poliglota. 1. ed. São Paulo: Novatec, 2015.

SCHWABER, K.; SUTHERLAND J. **Um guia definitivo para o *scrum***: as regras do jogo. ScrumInc, 2013. Disponível em: <<https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>>. Acesso em: 20/05/2017.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson, 2011.

SOUSA, M. **Desenvolvimento com *javaserver faces* – parte 1**. Rio de Janeiro: Java Magazine, p.8-18, 2015.

TEIXEIRA, M. **Estudo da utilização de *frameworks* no desenvolvimento de aplicações web**. Cascavel, 2008. Disponível em: <<http://www.inf.unioeste.br/~tcc/2008/TCC%20-%20Marcelo%20Teixeira.pdf>>. Acesso em: 24/05/2017

TOMASEL, Tiago Souza. **Aplicativo de reconhecimento de imagens em dispositivos móveis para ambientes previamente mapeados**. Trabalho de conclusão de curso de graduação em Ciência da Computação – Ciência da Computação do Centro Universitário La Salle – Unilasalle. Disponível em <http://biblioteca.unilasalle.edu.br/docs_online/tcc/graduacao/ciencia_da_computacao/2014/tstomasel.pdf>. Acessado em: 29 jan. 2016.

VIEIRA M. R.; FIGUEIREDO, J.M.; LIBERATTI, G.; VIEBRANTZ, A. F. M. **Bancos de Dados NoSQL**: Conceitos, Ferramentas, Linguagens e Estudos de Casos no Contexto de Big Data. SBBD, 2012. Disponível em: <http://data.ime.usp.br/sbbd2012/artigos/pdfs/sbbd_min_01.pdf>. Acesso em: 24/05/2017.

ANEXO A – Carta De Autorização



CARTA DE AUTORIZAÇÃO

Goiânia, 06 de Junho de 2017

Autorizamos o Sr. Daniel Ferreira Rabelo e o Sr. Marco Vinicius Isecke Cândido, a utilização do nome e dados estatísticos da empresa Gestão Consulting Services para serem utilizadas na pesquisa: Análise de Desempenho de Banco de Dados NoSQL em Sistemas com Baixo Volume de Dados.

Esta autorização está concedida aos pesquisadores, comprometendo-se os mesmos a utilizar os dados para fins científicos, garantindo a não utilização das informações em prejuízo da empresa.

Paulo Sousa
Gestão Consulting Services

ANEXO B – Testes de Carga

B.1. Inserção de Plano de Contas

1674 registros

```

WildFly 8.x [JBoss Application Server Startup Configuration] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (13 de nov de 2017 21:08:17)
21:09:27,477 INFO [org.mongodb.driver.cluster] (default task-28) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
21:09:27,482 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a262744607c24c098c846', description='null'}-localhost:27017) Opened connection [connectionId{localValue:7, serverValue:84}] to localhost:27017
21:09:27,482 INFO [org.mongodb.driver.cluster] (default task-28) No server chosen by WritableServerSelector from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{address=localhost:27017, type=UNKNOWN, state=CONNECTING}]}
21:09:27,482 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a262744607c24c098c846', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTING}
21:09:27,485 INFO [org.mongodb.driver.connection] (default task-28) Opened connection [connectionId{localValue:8, serverValue:85}] to localhost:27017
21:09:27,495 INFO [org.mongodb.driver.connection] (default task-28) Closed connection [connectionId{localValue:8, serverValue:85}] to localhost:27017 because the pool has been closed.
21:09:27,497 INFO [stdout] (default task-28) Mongo Período plano de contas: 19ms

21:09:27,512 INFO [stdout] (default task-28) Relacional Período plano de contas: 15ms

21:09:27,523 INFO [org.mongodb.driver.cluster] (default task-28) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
21:09:27,524 INFO [org.mongodb.driver.cluster] (default task-28) No server chosen by WritableServerSelector from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{address=localhost:27017, type=UNKNOWN, state=CONNECTING}]}
21:09:27,527 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a262744607c24c098c848', description='null'}-localhost:27017) Opened connection [connectionId{localValue:9, serverValue:86}] to localhost:27017
21:09:27,528 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a262744607c24c098c848', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTING}
21:09:27,530 INFO [org.mongodb.driver.connection] (default task-28) Opened connection [connectionId{localValue:10, serverValue:87}] to localhost:27017
21:09:27,678 INFO [org.mongodb.driver.connection] (default task-28) Closed connection [connectionId{localValue:10, serverValue:87}] to localhost:27017 because the pool has been closed.
21:09:27,679 INFO [stdout] (default task-28) Plano de contas Mongo: 156ms

21:09:29,054 INFO [stdout] (default task-28) Relacional Plano de contas: 1375ms

21:09:29,055 INFO [org.mongodb.driver.cluster] (default task-28) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
21:09:29,057 INFO [org.mongodb.driver.cluster] (default task-28) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{address=localhost:27017, type=UNKNOWN, state=CONNECTING}]}
21:09:29,059 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a262944607c24c098cc5a', description='null'}-localhost:27017) Opened connection [connectionId{localValue:11, serverValue:88}] to localhost:27017
21:09:29,060 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a262944607c24c098cc5a', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTING}
21:09:29,064 INFO [org.mongodb.driver.connection] (default task-28) Opened connection [connectionId{localValue:12, serverValue:89}] to localhost:27017
21:09:29,084 INFO [org.mongodb.driver.connection] (default task-28) Closed connection [connectionId{localValue:12, serverValue:89}] to localhost:27017 because the pool has been closed.
21:09:29,089 INFO [org.mongodb.driver.cluster] (default task-28) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
21:09:29,091 INFO [org.mongodb.driver.cluster] (default task-28) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{address=localhost:27017, type=UNKNOWN, state=CONNECTING}]}
21:09:29,093 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a262944607c24c098cc5b', description='null'}-localhost:27017) Opened connection [connectionId{localValue:13, serverValue:90}] to localhost:27017
21:09:29,093 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a262944607c24c098cc5b', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTING}
21:09:29,096 INFO [org.mongodb.driver.connection] (default task-28) Opened connection [connectionId{localValue:14, serverValue:91}] to localhost:27017
21:09:29,097 INFO [org.mongodb.driver.connection] (default task-28) Closed connection [connectionId{localValue:14, serverValue:91}] to localhost:27017 because the pool has been closed.
21:09:31,573 WARNING [javax.enterprise.resource.webcontainer.jsf.context] (default task-29) JSF1091: Nenhum tipo de mime pôde ser encontrado para o arquivo fa/fontawesome-webfont.woff2. Para resolver isso, adicione um mapeamento de mime-type ao web.xml do apl:
21:09:31,575 WARNING [javax.enterprise.resource.webcontainer.jsf.context] (default task-29) JSF1091: Nenhum tipo de mime pôde ser encontrado para o arquivo fa/fontawesome-webfont.woff2. Para resolver isso, adicione um mapeamento de mime-type ao web.xml do apl:
  
```

10.000 registros

```

SGC-JSF - Java EE - sdc/src/main/java/br/com/macro/business/PlanoContaBusiness.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help

WildFly 8.x [JBoss Application Server Startup Configuration] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (13 de nov de 2017 20:46:53)
20:55:10,380 INFO [org.mongodb.driver.cluster] (default task-36) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
20:55:10,387 INFO [org.mongodb.driver.cluster] (default task-36) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{address=localhost:27017, type=UNKN
20:55:10,388 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a22ce44607c1fd43ff7fd', description='null'}-localhost:27017) Opened connection [connectionId{localValue:7, serverValue:40}] to localhost:27017
20:55:10,389 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a22ce44607c1fd43ff7fd', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALON
20:55:10,393 INFO [org.mongodb.driver.connection] (default task-36) Opened connection [connectionId{localValue:8, serverValue:41}] to localhost:27017
20:55:10,404 INFO [org.mongodb.driver.connection] (default task-36) Closed connection [connectionId{localValue:8, serverValue:41}] to localhost:27017 because the pool has been closed.
20:55:10,406 INFO [stdout] (default task-36) Mongo Período plano de contas: 25ms

20:55:10,506 INFO [stdout] (default task-36) Relacional Período plano de contas: 100ms

20:55:10,521 INFO [org.mongodb.driver.cluster] (default task-36) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
20:55:10,523 INFO [org.mongodb.driver.cluster] (default task-36) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{address=localhost:27017, type=UNKN
20:55:10,527 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a22ce44607c1fd43ff7fd', description='null'}-localhost:27017) Opened connection [connectionId{localValue:9, serverValue:42}] to localhost:27017
20:55:10,527 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a22ce44607c1fd43ff7fd', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALON
20:55:10,530 INFO [org.mongodb.driver.connection] (default task-36) Opened connection [connectionId{localValue:10, serverValue:43}] to localhost:27017
20:55:11,397 INFO [org.mongodb.driver.connection] (default task-36) Closed connection [connectionId{localValue:10, serverValue:43}] to localhost:27017 because the pool has been closed.
20:55:11,399 INFO [stdout] (default task-36) Plano de contas Mongo: 87ms

20:56:13,107 INFO [stdout] (default task-36) Relacional Plano de contas: 61707ms

20:56:13,108 INFO [org.mongodb.driver.cluster] (default task-36) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
20:56:13,111 INFO [org.mongodb.driver.cluster] (default task-36) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{ad
20:56:13,148 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a230d44607c1fd4402109', description='null'}-localhost:27017) Opened connection [connectionId{localValue:11, serverValue:44}] to localhost:27017
20:56:13,149 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a230d44607c1fd4402109', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALON
20:56:13,177 INFO [org.mongodb.driver.connection] (default task-36) Opened connection [connectionId{localValue:12, serverValue:45}] to localhost:27017
20:56:13,292 INFO [org.mongodb.driver.connection] (default task-36) Closed connection [connectionId{localValue:12, serverValue:45}] to localhost:27017 because the pool has been closed.
20:56:13,440 INFO [org.mongodb.driver.cluster] (default task-36) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
20:56:13,441 INFO [org.mongodb.driver.cluster] (default task-36) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{ad
20:56:13,444 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a230d44607c1fd440210a', description='null'}-localhost:27017) Opened connection [connectionId{localValue:13, serverValue:46}] to localhost:27017
20:56:13,445 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a230d44607c1fd440210a', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALON
20:56:13,449 INFO [org.mongodb.driver.connection] (default task-36) Opened connection [connectionId{localValue:14, serverValue:47}] to localhost:27017
20:56:13,450 INFO [org.mongodb.driver.connection] (default task-36) Closed connection [connectionId{localValue:14, serverValue:47}] to localhost:27017 because the pool has been closed.

```

100.000 registros

SGC-JSF - Java EE - sdc/src/main/java/br/com/macro/persistence/GenericDAO.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Markers Properties Servers Console Data Source Explorer Snippets Progress Search Debug History JUnit

WildFly 8.x [JBoss Application Server Startup Configuration] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (13 de nov de 2017 23:30:31)

```

23:31:59,003 INFO [org.mongodb.driver.cluster] (default task-38) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
23:31:59,010 INFO [org.mongodb.driver.cluster] (default task-38) No server chosen by WritableServerSelector from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{address=localhost:27017, type=UN
23:31:59,011 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a478f44607c333cb1e1b0', description='null'}-localhost:27017) Opened connection [connectionId{localValue:11, serverValue:300}] to localhost:27017
23:31:59,011 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a478f44607c333cb1e1b0', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALON
23:31:59,014 INFO [org.mongodb.driver.connection] (default task-38) Opened connection [connectionId{localValue:12, serverValue:301}] to localhost:27017
23:31:59,020 INFO [org.mongodb.driver.connection] (default task-38) Closed connection [connectionId{localValue:12, serverValue:301}] to localhost:27017 because the pool has been closed.
23:31:59,020 INFO [stdout] (default task-38) Mongo Período plano de contas: 17ms

23:31:59,042 INFO [stdout] (default task-38) Relacional Período plano de contas: 22ms

23:31:59,167 INFO [org.mongodb.driver.cluster] (default task-38) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
23:31:59,169 INFO [org.mongodb.driver.cluster] (default task-38) No server chosen by WritableServerSelector from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{address=localhost:27017, type=UN
23:31:59,197 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a478f44607c333cb1e1b2', description='null'}-localhost:27017) Opened connection [connectionId{localValue:13, serverValue:302}] to localhost:27017
23:31:59,198 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a478f44607c333cb1e1b2', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALON
23:31:59,201 INFO [org.mongodb.driver.connection] (default task-38) Opened connection [connectionId{localValue:14, serverValue:303}] to localhost:27017
23:32:06,246 INFO [org.mongodb.driver.connection] (default task-38) Closed connection [connectionId{localValue:14, serverValue:303}] to localhost:27017 because the pool has been closed.
23:32:06,278 INFO [stdout] (default task-38) Plano de contas Mongo: 7112ms

23:32:21,048 INFO [stdout] (default task-38) Relacional Plano de contas: 14770ms

23:32:21,049 INFO [org.mongodb.driver.cluster] (default task-38) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
23:32:21,050 INFO [org.mongodb.driver.cluster] (default task-38) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{ad
23:32:21,071 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a47a544607c333cb375d2', description='null'}-localhost:27017) Opened connection [connectionId{localValue:15, serverValue:304}] to localhost:27017
23:32:21,071 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a47a544607c333cb375d2', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALON
23:32:21,074 INFO [org.mongodb.driver.connection] (default task-38) Opened connection [connectionId{localValue:16, serverValue:305}] to localhost:27017
23:32:21,075 INFO [org.mongodb.driver.connection] (default task-38) Closed connection [connectionId{localValue:16, serverValue:305}] to localhost:27017 because the pool has been closed.
23:32:21,197 INFO [org.mongodb.driver.cluster] (default task-38) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
23:32:21,201 INFO [org.mongodb.driver.cluster] (default task-38) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{ad
23:32:21,210 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a47a544607c333cb375d3', description='null'}-localhost:27017) Opened connection [connectionId{localValue:17, serverValue:306}] to localhost:27017
23:32:21,211 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a47a544607c333cb375d3', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALON
23:32:21,222 INFO [org.mongodb.driver.connection] (default task-38) Opened connection [connectionId{localValue:18, serverValue:307}] to localhost:27017
23:32:21,225 INFO [org.mongodb.driver.connection] (default task-38) Closed connection [connectionId{localValue:18, serverValue:307}] to localhost:27017 because the pool has been closed.

```


B.2. Visualizar de Plano de Contas

1674 registros

```

SGC-JSF - Java EE - sdc/src/main/java/br/com/macro/business/PlanoContaBusiness.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help

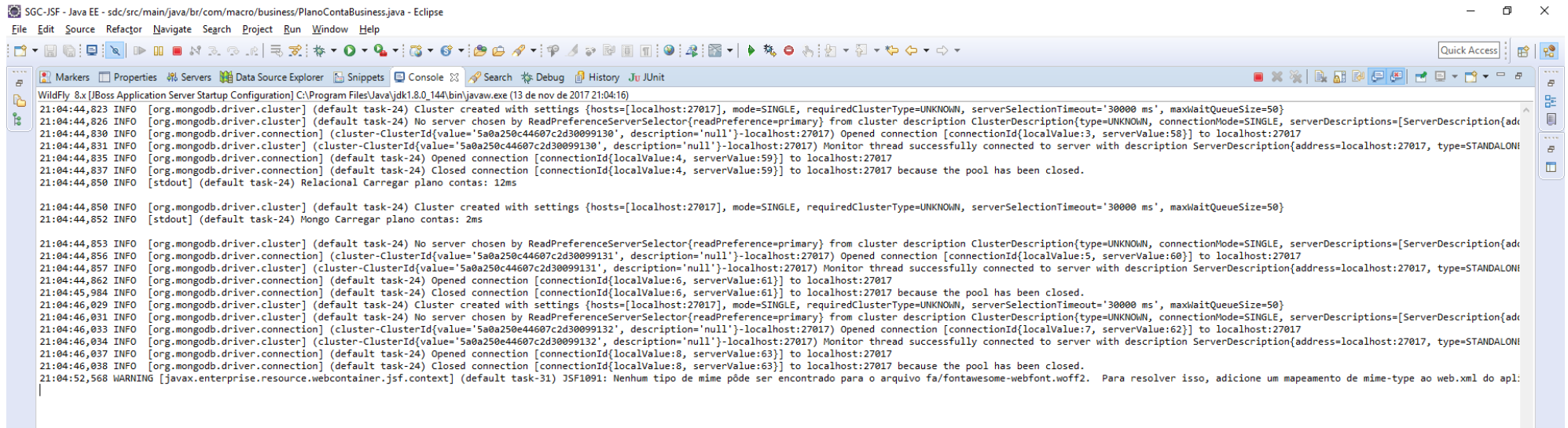
Markers Properties Servers Data Source Explorer Snippets Console Search Debug History JUnit
WildFly 8.x [JBoss Application Server Startup Configuration] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (13 de nov de 2017 21:08:17)
21:10:25,925 WARNING [javax.enterprise.resource.webcontainer.jsf.context] (default task-36) JSF1091: Nenhum tipo de mime pôde ser encontrado para o arquivo fa/fontawesome-webfont.woff2. Para resolver isso, adicione um mapeamento de mime-type ao web.xml do apl:
21:10:27,184 INFO [org.mongodb.driver.cluster] (default task-38) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
21:10:27,188 INFO [org.mongodb.driver.cluster] (default task-38) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{ad
21:10:27,191 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a266344607c24c098cc5c', description='null'}-localhost:27017) Opened connection [connectionId{localValue:15, serverValue:92}] to localhost:27017
21:10:27,191 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a266344607c24c098cc5c', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALON
21:10:27,194 INFO [org.mongodb.driver.connection] (default task-38) Opened connection [connectionId{localValue:16, serverValue:93}] to localhost:27017
21:10:27,196 INFO [org.mongodb.driver.connection] (default task-38) Closed connection [connectionId{localValue:16, serverValue:93}] to localhost:27017 because the pool has been closed.
21:10:27,204 INFO [stdout] (default task-38) Relacional Carregar plano contas: 7ms

21:10:27,205 INFO [org.mongodb.driver.cluster] (default task-38) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
21:10:27,206 INFO [stdout] (default task-38) Mongo Carregar plano contas: 2ms

21:10:27,207 INFO [org.mongodb.driver.cluster] (default task-38) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{ad
21:10:27,208 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a266344607c24c098cc5d', description='null'}-localhost:27017) Opened connection [connectionId{localValue:17, serverValue:94}] to localhost:27017
21:10:27,209 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a266344607c24c098cc5d', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALON
21:10:27,212 INFO [org.mongodb.driver.connection] (default task-38) Opened connection [connectionId{localValue:18, serverValue:95}] to localhost:27017
21:10:27,448 INFO [org.mongodb.driver.connection] (default task-38) Closed connection [connectionId{localValue:18, serverValue:95}] to localhost:27017 because the pool has been closed.
21:10:27,453 INFO [org.mongodb.driver.cluster] (default task-38) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
21:10:27,455 INFO [org.mongodb.driver.cluster] (default task-38) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{ad
21:10:27,457 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a266344607c24c098cc5e', description='null'}-localhost:27017) Opened connection [connectionId{localValue:19, serverValue:96}] to localhost:27017
21:10:27,457 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a266344607c24c098cc5e', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALON
21:10:27,460 INFO [org.mongodb.driver.connection] (default task-38) Opened connection [connectionId{localValue:20, serverValue:97}] to localhost:27017
21:10:27,461 INFO [org.mongodb.driver.connection] (default task-38) Closed connection [connectionId{localValue:20, serverValue:97}] to localhost:27017 because the pool has been closed.
21:10:28,389 WARNING [javax.enterprise.resource.webcontainer.jsf.context] (default task-45) JSF1091: Nenhum tipo de mime pôde ser encontrado para o arquivo fa/fontawesome-webfont.woff2. Para resolver isso, adicione um mapeamento de mime-type ao web.xml do apl:

```

10.000 registros



```
SGC-JSF - Java EE - sdc/src/main/java/br/com/macro/business/PlanoContaBusiness.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help

WildFly 8.x [JBoss Application Server Startup Configuration] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (13 de nov de 2017 21:04:16)
21:04:44,823 INFO [org.mongodb.driver.cluster] (default task-24) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
21:04:44,826 INFO [org.mongodb.driver.cluster] (default task-24) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{ad
21:04:44,830 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a250c44607c2d30099130', description='null'}-localhost:27017) Opened connection [connectionId{localValue:3, serverValue:58}] to localhost:27017
21:04:44,831 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a250c44607c2d30099130', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE
21:04:44,835 INFO [org.mongodb.driver.connection] (default task-24) Opened connection [connectionId{localValue:4, serverValue:59}] to localhost:27017
21:04:44,837 INFO [org.mongodb.driver.connection] (default task-24) Closed connection [connectionId{localValue:4, serverValue:59}] to localhost:27017 because the pool has been closed.
21:04:44,850 INFO [stdout] (default task-24) Relacional Carregar plano contas: 12ms

21:04:44,850 INFO [org.mongodb.driver.cluster] (default task-24) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
21:04:44,852 INFO [stdout] (default task-24) Mongo Carregar plano contas: 2ms

21:04:44,853 INFO [org.mongodb.driver.cluster] (default task-24) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{ad
21:04:44,856 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a250c44607c2d30099131', description='null'}-localhost:27017) Opened connection [connectionId{localValue:5, serverValue:60}] to localhost:27017
21:04:44,857 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a250c44607c2d30099131', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE
21:04:44,862 INFO [org.mongodb.driver.connection] (default task-24) Opened connection [connectionId{localValue:6, serverValue:61}] to localhost:27017
21:04:45,904 INFO [org.mongodb.driver.connection] (default task-24) Closed connection [connectionId{localValue:6, serverValue:61}] to localhost:27017 because the pool has been closed.
21:04:46,029 INFO [org.mongodb.driver.cluster] (default task-24) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
21:04:46,031 INFO [org.mongodb.driver.cluster] (default task-24) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{ad
21:04:46,033 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a250c44607c2d30099132', description='null'}-localhost:27017) Opened connection [connectionId{localValue:7, serverValue:62}] to localhost:27017
21:04:46,034 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a250c44607c2d30099132', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE
21:04:46,037 INFO [org.mongodb.driver.connection] (default task-24) Opened connection [connectionId{localValue:8, serverValue:63}] to localhost:27017
21:04:46,038 INFO [org.mongodb.driver.connection] (default task-24) Closed connection [connectionId{localValue:8, serverValue:63}] to localhost:27017 because the pool has been closed.
21:04:52,568 WARNING [javax.enterprise.resource.webcontainer.jsf.context] (default task-31) JSF1091: Nenhum tipo de mime pôde ser encontrado para o arquivo fa/fontawesome-webfont.woff2. Para resolver isso, adicione um mapeamento de mime-type ao web.xml do apl:
```

100.000 registros

```

SGC-JSF - Java EE - sdc/src/main/java/br/com/macro/persistence/GenericDAO.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Markers Properties Servers Console Data Source Explorer Snippets Progress Search Debug History JUnit
WildFly 8x [Boss: Application Server Startup Configuration] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (13 de nov de 2017 23:38:38)
23:39:56,283 INFO [org.mongodb.driver.cluster] (default task-15) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
23:39:56,286 INFO [org.mongodb.driver.cluster] (default task-15) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{address=localhost:27017, type=STANDALONE}}]
23:39:56,291 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a496c44607c26a458a99e', description='null'}-localhost:27017) Opened connection [connectionId{localValue:3, serverValue:324}] to localhost:27017
23:39:56,292 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a496c44607c26a458a99e', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE}
23:39:56,297 INFO [org.mongodb.driver.connection] (default task-15) Opened connection [connectionId{localValue:4, serverValue:325}] to localhost:27017
23:39:56,298 INFO [org.mongodb.driver.connection] (default task-15) Closed connection [connectionId{localValue:4, serverValue:325}] to localhost:27017 because the pool has been closed.
23:39:56,343 INFO [stdout] (default task-15) Relacional Carregar plano contas: 43ms
23:39:56,343 INFO [org.mongodb.driver.cluster] (default task-15) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
23:39:56,347 INFO [stdout] (default task-15) Mongo Carregar plano contas: 3ms
23:39:56,347 INFO [org.mongodb.driver.cluster] (default task-15) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{address=localhost:27017, type=STANDALONE}}]
23:39:56,369 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a496c44607c26a458a99f', description='null'}-localhost:27017) Opened connection [connectionId{localValue:5, serverValue:326}] to localhost:27017
23:39:56,370 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a496c44607c26a458a99f', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE}
23:39:56,378 INFO [org.mongodb.driver.connection] (default task-15) Opened connection [connectionId{localValue:6, serverValue:327}] to localhost:27017
23:40:02,627 INFO [org.mongodb.driver.connection] (default task-15) Closed connection [connectionId{localValue:6, serverValue:327}] to localhost:27017 because the pool has been closed.
23:40:02,750 INFO [org.mongodb.driver.cluster] (default task-15) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
23:40:02,755 INFO [org.mongodb.driver.cluster] (default task-15) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{address=localhost:27017, type=STANDALONE}}]
23:40:02,764 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a497244607c26a458a9a0', description='null'}-localhost:27017) Opened connection [connectionId{localValue:7, serverValue:328}] to localhost:27017
23:40:02,766 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a497244607c26a458a9a0', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE}
23:40:02,775 INFO [org.mongodb.driver.connection] (default task-15) Opened connection [connectionId{localValue:8, serverValue:329}] to localhost:27017
23:40:02,777 INFO [org.mongodb.driver.connection] (default task-15) Closed connection [connectionId{localValue:8, serverValue:329}] to localhost:27017 because the pool has been closed.

```

B.3. Importar Balancete

1674 registros

The screenshot shows the Eclipse IDE console with the following log entries:

```

WildFly 8.x [JBoss Application Server Startup Configuration] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (13 de nov de 2017 23:43:30)
23:45:20,009 INFO [org.mongodb.driver.connection] (default task-60) Closed connection [connectionId{localValue:18, serverValue:347}] to localhost:27017 because the pool has been closed.
23:45:20,048 INFO [stdout] (default task-60) Relacional Carregar plano contas: 37ms

23:45:20,109 INFO [org.mongodb.driver.cluster] (default task-60) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
23:45:20,111 INFO [org.mongodb.driver.cluster] (default task-60) No server chosen by WritableServerSelector from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{address=localhost:27017, type=UN
23:45:20,115 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a4ab044607c0dbc64a756', description='null'}-localhost:27017) Opened connection [connectionId{localValue:19, serverValue:348}] to localhost:27017
23:45:20,116 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a4ab044607c0dbc64a756', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALON
23:45:20,120 INFO [org.mongodb.driver.connection] (default task-60) Opened connection [connectionId{localValue:20, serverValue:349}] to localhost:27017
23:45:20,162 INFO [org.mongodb.driver.connection] (default task-60) Closed connection [connectionId{localValue:20, serverValue:349}] to localhost:27017 because the pool has been closed.
23:45:20,163 INFO [stdout] (default task-60) Mongo (Salvar Balancete): 54ms

23:45:20,315 INFO [stdout] (default task-60) Relacional (Salvar Balancete): 147ms
  
```

10.000 registros

The screenshot shows the Eclipse IDE console with the following log entries:

```

WildFly 8.x [JBoss Application Server Startup Configuration] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (13 de nov de 2017 23:43:30)
23:47:08,996 INFO [org.mongodb.driver.connection] (default task-63) Closed connection [connectionId{localValue:24, serverValue:353}] to localhost:27017 because the pool has been closed.
23:47:09,032 INFO [stdout] (default task-63) Relacional Carregar plano contas: 35ms

23:47:34,182 INFO [org.mongodb.driver.cluster] (default task-63) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
23:47:34,185 INFO [org.mongodb.driver.cluster] (default task-63) No server chosen by WritableServerSelector from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{address=localhost:27017, type=UN
23:47:34,204 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a4b3644607c0dbc64a930', description='null'}-localhost:27017) Opened connection [connectionId{localValue:25, serverValue:354}] to localhost:27017
23:47:34,205 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a4b3644607c0dbc64a930', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALON
23:47:34,207 INFO [org.mongodb.driver.connection] (default task-63) Opened connection [connectionId{localValue:26, serverValue:355}] to localhost:27017
23:47:34,596 INFO [org.mongodb.driver.connection] (default task-63) Closed connection [connectionId{localValue:26, serverValue:355}] to localhost:27017 because the pool has been closed.
23:47:34,669 INFO [stdout] (default task-63) Mongo (Salvar Balancete): 487ms

23:47:36,099 INFO [stdout] (default task-63) Relacional (Salvar Balancete): 1427ms
  
```

100.000 registros

```

SGC-JSF - Java EE - sdc/src/main/java/br/com/macro/business/BalanceteBusiness.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help

WildFly 8.x [JBoss Application Server Startup Configuration] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (13 de nov de 2017 23:43:30)
23:50:47,926 INFO [org.mongodb.driver.connection] (default task-67) Closed connection [connectionId{localValue:36, serverValue:365}] to localhost:27017 because the pool has been closed.
23:50:47,969 INFO [stdout] (default task-67) Relacional Carregar plano contas: 42ms

23:55:22,397 INFO [org.mongodb.driver.cluster] (default task-67) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
23:55:22,399 INFO [org.mongodb.driver.cluster] (default task-67) No server chosen by WritableServerSelector from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{address=localhost:27017, type=UN}
23:55:22,418 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0a4d0a44607c0dbbc64dcf6', description='null'}-localhost:27017) Opened connection [connectionId{localValue:37, serverValue:366}] to localhost:27017
23:55:22,419 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0a4d0a44607c0dbbc64dcf6', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALON
23:55:22,421 INFO [org.mongodb.driver.connection] (default task-67) Opened connection [connectionId{localValue:38, serverValue:367}] to localhost:27017
23:55:27,151 INFO [org.mongodb.driver.connection] (default task-67) Closed connection [connectionId{localValue:38, serverValue:367}] to localhost:27017 because the pool has been closed.
23:55:27,183 INFO [stdout] (default task-67) Mongo (Salvar Balancete): 4787ms

23:55:37,878 INFO [stdout] (default task-67) Relacional (Salvar Balancete): 10692ms
  
```

B.4. Relatório de Balanço Patrimonial

1674 registros

```

SGC-JSF - Java EE - sdc/src/main/java/br/com/macro/controller/RelContabilController.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help

WildFly 8.x [JBoss Application Server Startup Configuration] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (15 de nov de 2017 13:44:55)
13:48:28,601 INFO [org.mongodb.driver.cluster] (default task-51) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
13:48:28,602 INFO [stdout] (default task-51) Mongo Carregar plano contas: 1ms

13:48:28,603 INFO [org.mongodb.driver.cluster] (default task-51) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{ad
13:48:28,605 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0c61cc44607c349c1a8f0f', description='null'}-localhost:27017) Opened connection [connectionId{localValue:43, serverValue:514}] to localhost:27017
13:48:28,606 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0c61cc44607c349c1a8f0f', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALON
13:48:28,613 INFO [org.mongodb.driver.connection] (default task-51) Opened connection [connectionId{localValue:44, serverValue:515}] to localhost:27017
13:48:28,672 INFO [org.mongodb.driver.connection] (default task-51) Closed connection [connectionId{localValue:44, serverValue:515}] to localhost:27017 because the pool has been closed.
13:48:28,675 INFO [stdout] (default task-51) Relacional Carregar plano contas: 2ms
  
```

10.000 registros

SGC-JSF - Java EE - sdc/src/main/java/br/com/macro/controller/RelContabilController.java - Eclipse

```

WildFly 8.x [JBoss Application Server Startup Configuration] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (15 de nov de 2017 13:50:11)
13:57:36,821 INFO [org.mongodb.driver.cluster] (default task-65) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
13:57:36,823 INFO [stdout] (default task-65) Mongo Carregar plano contas: 1ms

13:57:36,823 INFO [org.mongodb.driver.cluster] (default task-65) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{ad
13:57:36,826 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0c63f044607c4388149214', description='null'}-localhost:27017) Opened connection [connectionId{localValue:51, serverValue:596}] to localhost:27017
13:57:36,826 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0c63f044607c4388149214', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE
13:57:36,830 INFO [org.mongodb.driver.connection] (default task-65) Opened connection [connectionId{localValue:52, serverValue:597}] to localhost:27017
13:57:37,125 INFO [org.mongodb.driver.connection] (default task-65) Closed connection [connectionId{localValue:52, serverValue:597}] to localhost:27017 because the pool has been closed.
13:57:37,130 INFO [stdout] (default task-65) Relacional          Carregar plano contas: 4ms
  
```

100.000 registros

SGC-JSF - Java EE - sdc/src/main/java/br/com/macro/business/PlanoContaBusiness.java - Eclipse

```

WildFly 8.x [JBoss Application Server Startup Configuration] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (15 de nov de 2017 14:19:02)
14:25:59,632 INFO [org.mongodb.driver.connection] (default task-54) Closed connection [connectionId{localValue:32, serverValue:649}] to localhost:27017 because the pool has been closed.
14:25:59,637 INFO [org.mongodb.driver.cluster] (default task-54) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
14:25:59,639 INFO [org.mongodb.driver.cluster] (default task-54) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{ad
14:25:59,641 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0c6a9744607c1af0df4e05', description='null'}-localhost:27017) Opened connection [connectionId{localValue:33, serverValue:650}] to localhost:27017
14:25:59,642 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0c6a9744607c1af0df4e05', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE
14:25:59,645 INFO [org.mongodb.driver.connection] (default task-54) Opened connection [connectionId{localValue:34, serverValue:651}] to localhost:27017
14:25:59,646 INFO [org.mongodb.driver.connection] (default task-54) Closed connection [connectionId{localValue:34, serverValue:651}] to localhost:27017 because the pool has been closed.
14:25:59,869 WARNING [javax.enterprise.resource.webcontainer.jsf.context] (default task-59) JSF1891: Nenhum tipo de mime pôde ser encontrado para o arquivo fa/fontawesome-webfont.woff2. Para resolver isso, adicione um mapeamento de mime-type ao web.xml do apl:
14:26:03,828 INFO [org.mongodb.driver.cluster] (default task-61) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
14:26:03,830 INFO [stdout] (default task-61) Mongo Carregar plano contas: 2ms

14:26:03,831 INFO [org.mongodb.driver.cluster] (default task-61) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{ad
14:26:03,834 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0c6a9b44607c1af0df4e06', description='null'}-localhost:27017) Opened connection [connectionId{localValue:35, serverValue:652}] to localhost:27017
14:26:03,835 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0c6a9b44607c1af0df4e06', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE
14:26:03,838 INFO [org.mongodb.driver.connection] (default task-61) Opened connection [connectionId{localValue:36, serverValue:653}] to localhost:27017
14:26:17,460 INFO [org.mongodb.driver.connection] (default task-61) Closed connection [connectionId{localValue:36, serverValue:653}] to localhost:27017 because the pool has been closed.
14:26:17,484 INFO [stdout] (default task-61) Relacional          Carregar plano contas: 23ms
  
```

B.5. Relatório de Demonstração de Fluxo de Caixa

1674 registros

The screenshot shows the Eclipse IDE console with the following log output:

```

SGC-JSF - Java EE - sdc/src/main/java/br/com/macro/controller/RelPlanoContasController.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Markers Properties Servers Console Data Source Explorer Snippets Progress Search Debug History JUnit
WildFly 8.x [JBoss Application Server Startup Configuration] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (15 de nov de 2017 14:34:08)
14:35:05,300 INFO [org.mongodb.driver.cluster] (default task-32) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
14:35:05,302 INFO [stdout] (default task-32) Mongo Carregar plano contas: 1ms
14:35:05,302 INFO [org.mongodb.driver.cluster] (default task-32) No server chosen by ReadPreferenceServerSelector(readPreference=primary) from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{ad
14:35:05,306 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0c6cb944607c2a68977daa', description='null'}-localhost:27017) Opened connection [connectionId{localValue:23, serverValue:726}] to localhost:27017
14:35:05,306 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0c6cb944607c2a68977daa', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALON
14:35:05,309 INFO [org.mongodb.driver.connection] (default task-32) Opened connection [connectionId{localValue:24, serverValue:727}] to localhost:27017
14:35:05,308 INFO [org.mongodb.driver.connection] (default task-32) Closed connection [connectionId{localValue:24, serverValue:727}] to localhost:27017 because the pool has been closed.
14:35:05,384 INFO [stdout] (default task-32) Relacional Carregar plano contas: 4ms
  
```

10.000 registros

The screenshot shows the Eclipse IDE console with the following log output:

```

SGC-JSF - Java EE - sdc/src/main/java/br/com/macro/controller/RelContabilController.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Markers Properties Servers Console Data Source Explorer Snippets Progress Search Debug History JUnit
WildFly 8.x [JBoss Application Server Startup Configuration] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (15 de nov de 2017 13:50:11)
13:56:50,914 INFO [org.mongodb.driver.cluster] (default task-45) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
13:56:50,915 INFO [stdout] (default task-45) Mongo Carregar plano contas: 1ms
13:56:50,915 INFO [org.mongodb.driver.cluster] (default task-45) No server chosen by ReadPreferenceServerSelector(readPreference=primary) from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{ad
13:56:50,920 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0c63c244607c438814920b', description='null'}-localhost:27017) Opened connection [connectionId{localValue:33, serverValue:578}] to localhost:27017
13:56:50,921 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0c63c244607c438814920b', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALON
13:56:50,927 INFO [org.mongodb.driver.connection] (default task-45) Opened connection [connectionId{localValue:34, serverValue:579}] to localhost:27017
13:56:51,211 INFO [org.mongodb.driver.connection] (default task-45) Closed connection [connectionId{localValue:34, serverValue:579}] to localhost:27017 because the pool has been closed.
13:56:51,216 INFO [stdout] (default task-45) Relacional Carregar plano contas: 5ms
  
```

100.000 registros

```

SGC-JSF - Java EE - sdc/src/main/java/br/com/macro/controller/RelPlanoContasController.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Markers Properties Servers Console Data Source Explorer Snippets Progress Search Debug History JUnit
WildFly 8.x [JBoss Application Server Startup Configuration] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (15 de nov de 2017 14:31:51)
14:33:09,911 INFO [org.mongodb.driver.cluster] (default task-63) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
14:33:09,916 INFO [stdout] (default task-63) Mongo Carregar plano contas: 5ms
14:33:09,917 INFO [org.mongodb.driver.cluster] (default task-63) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{ad
14:33:09,925 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0c6c4544607c228099c2b3', description='null'}-localhost:27017) Opened connection [connectionId{localValue:17, serverValue:702}] to localhost:27017
14:33:09,926 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0c6c4544607c228099c2b3', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE
14:33:09,937 INFO [org.mongodb.driver.connection] (default task-63) Opened connection [connectionId{localValue:18, serverValue:703}] to localhost:27017
14:33:23,795 INFO [org.mongodb.driver.connection] (default task-63) Closed connection [connectionId{localValue:18, serverValue:703}] to localhost:27017 because the pool has been closed.
14:33:23,850 INFO [stdout] (default task-63) Relacional Carregar plano contas: 23ms

```

B.6. Relatório de Demonstração de Mutaç o do Patrim nio L quido

1674 registros

```

SGC-JSF - Java EE - sdc/src/main/java/br/com/macro/controller/RelPlanoContasController.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Markers Properties Servers Console Data Source Explorer Snippets Progress Search Debug History JUnit
WildFly 8.x [JBoss Application Server Startup Configuration] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (15 de nov de 2017 14:34:08)
14:36:29,712 INFO [org.mongodb.driver.cluster] (default task-48) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
14:36:29,713 INFO [stdout] (default task-48) Mongo Carregar balancete: 1ms
14:36:29,714 INFO [org.mongodb.driver.cluster] (default task-48) No server chosen by ReadPreferenceServerSelector{readPreference=primary} from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{ad
14:36:29,720 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0c6d0d44607c2a68977f89', description='null'}-localhost:27017) Opened connection [connectionId{localValue:39, serverValue:742}] to localhost:27017
14:36:29,720 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0c6d0d44607c2a68977f89', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE
14:36:29,724 INFO [org.mongodb.driver.connection] (default task-48) Opened connection [connectionId{localValue:40, serverValue:743}] to localhost:27017
14:36:29,730 INFO [org.mongodb.driver.connection] (default task-48) Closed connection [connectionId{localValue:40, serverValue:743}] to localhost:27017 because the pool has been closed.
14:36:29,738 INFO [stdout] (default task-48) Relacional Carregar balancete: 6ms

```

10.000 registros

SGC-JSF - Java EE - sdc/src/main/java/br/com/macro/controller/RelPlanoContasController.java - Eclipse

WildFly 8.x [JBoss Application Server Startup Configuration] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (15 de nov de 2017 14:37:03)

```

14:38:36,418 INFO [org.mongodb.driver.cluster] (default task-63) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
14:38:36,421 INFO [stdout] (default task-63) Mongo Carregar balancete: 2ms

14:38:36,422 INFO [org.mongodb.driver.cluster] (default task-63) No server chosen by ReadPreferenceServerSelector(readPreference=primary) from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{ad
14:38:36,428 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0c6d8c44607c2a4867b334', description='null'}-localhost:27017) Opened connection [connectionId{localValue:29, serverValue:772}] to localhost:27017
14:38:36,429 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0c6d8c44607c2a4867b334', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALON
14:38:36,433 INFO [org.mongodb.driver.connection] (default task-63) Opened connection [connectionId{localValue:30, serverValue:773}] to localhost:27017
14:38:36,538 INFO [org.mongodb.driver.connection] (default task-63) Closed connection [connectionId{localValue:30, serverValue:773}] to localhost:27017 because the pool has been closed.
14:38:36,558 INFO [stdout] (default task-63) Relacional          Carregar balancete: 19ms

```

100.000 registros

SGC-JSF - Java EE - sdc/src/main/java/br/com/macro/controller/RelPlanoContasController.java - Eclipse

WildFly 8.x [JBoss Application Server Startup Configuration] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (15 de nov de 2017 14:46:52)

```

14:48:49,388 INFO [org.mongodb.driver.cluster] (default task-26) Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=50}
14:48:49,390 INFO [stdout] (default task-26) Mongo Carregar balancete: 3ms

14:48:49,390 INFO [org.mongodb.driver.cluster] (default task-26) No server chosen by ReadPreferenceServerSelector(readPreference=primary) from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGLE, serverDescriptions=[ServerDescription{ad
14:48:49,393 INFO [org.mongodb.driver.connection] (cluster-ClusterId{value='5a0c6ff144607c24dc9a36a2', description='null'}-localhost:27017) Opened connection [connectionId{localValue:11, serverValue:834}] to localhost:27017
14:48:49,394 INFO [org.mongodb.driver.cluster] (cluster-ClusterId{value='5a0c6ff144607c24dc9a36a2', description='null'}-localhost:27017) Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALON
14:48:49,398 INFO [org.mongodb.driver.connection] (default task-26) Opened connection [connectionId{localValue:12, serverValue:835}] to localhost:27017
14:48:50,981 INFO [org.mongodb.driver.connection] (default task-26) Closed connection [connectionId{localValue:12, serverValue:835}] to localhost:27017 because the pool has been closed.
14:48:51,021 INFO [stdout] (default task-26) Relacional          Carregar balancete: 36ms

```

ANEXO C – Estórias de Usuário

B1. – EST01. Manter Usuário

SDC - SISTEMA DE DEMONSTRAÇÕES CONTÁBEIS

QUADRO DE TAREFAS

HISTÓRIA DE USUÁRIO

▲
0
#1 EST01 - Manter Usuário
>

release 1 ✕
sprint 1 ✕
+ Adicionar tag

Criado por Daniel Rabelo
17 fev 2017 22:46

Eu como administrador do sistema preciso cadastrar, alterar, ativar, inativar e/ou consultar usuários do sistema, para que possa realizar as demais funções do sistema.

Teste de Aceitação:

- Buscar Usuários -

Dado que o usuário possui permissão de administrador, quando acionar a opção "usuários", então poderá inserir, alterar, ativar, inativar ou buscar um usuário do sistema.

Dado que o usuário está na lista de usuários, quando optar buscar um usuário, então poderá filtrar os usuários conforme protótipo em anexo.

Dado que o usuário preencheu o formulário de filtro, quando acionar a opção "limpar", então o sistema limpa os campos de filtro.

- Inserção -

Dado que o usuário está na lista de usuários, quando acionado a opção "novo usuário", então poderá inserir um novo usuário.

Dado que o usuário está preenchendo o formulário de cadastro, quando informar um CPF inválido, então o sistema apresenta a mensagem de erro: "O CPF informado é inválido, tente outro."

Dado que o usuário está preenchendo o formulário de cadastro, quando informar um e-mail que não contenha no mínimo um "@" e um ".", então o sistema apresenta a mensagem de erro: "O e-mail deverá conter no mínimo um "@" e um "."!"

Dado que o usuário está preenchendo o campo de CEP, quando não souber o CEP, então poderá acionar a opção de "buscar CEP".

Dado que o usuário acionou a opção "buscar CEP", quando não souber o CEP, então o sistema abre em uma nova aba o link de busca de CEP do correios (buscacep.correios.com.br/sistemas/buscacep).

Dado que o usuário está preenchendo o campo de CEP, quando informar um CEP inválido, então o sistema apresenta a mensagem de erro: "O CEP informado é inexistente, tente outro.";

Dado que o usuário está preenchendo o campo de CEP, quando informar um CEP válido, então o sistema preenche automaticamente os campos: Cidade, Estado, Bairro (quando houver), Rua (quando houver);

Dado que o usuário preencheu o formulário de cadastrado, quando deixar um campo obrigatório em branco e acionar "salvar", então o sistema apresenta a mensagem de erro: "Os seguintes campos obrigatórios estão em branco: < campo1 >, < campo2 > e < campo3 >."

Dado que o usuário está preenchendo o formulário de cadastro, quando acionar a opção "cancelar", então o sistema retorna para a página anteriormente executada.

Dado que o usuário preencheu corretamente o formulário de cadastro, quando acionar a opção "inserir", então o sistema insere um novo usuário, atribui as permissões selecionadas para o usuário, e envia o e-mail de sucesso para o e-mail informado e apresenta a mensagem de sucesso: "O usuário < nome do novo usuário > foi inserido com sucesso!"

Dado que o usuário inseriu um novo usuário, quando acionar "inserir", então o sistema envia um e-mail de confirmação conforme o protótipo em anexo.

Dado que o sistema enviou o e-mail de confirmação, quando inseriu um novo usuário, então o sistema verifica se o e-mail foi enviado.

Dado que o sistema verifica que não foi enviado o e-mail, quando o usuário solicitou um novo cadastro, então o sistema insere o novo usuário e apresenta a mensagem de erro: "O usuário < nome do usuário > foi inserido, porém o e-mail de confirmação não pode ser enviado, tentaremos reenviar o e-mail novamente mais tarde".

Dado que o sistema verifica que não foi enviado o e-mail de confirmação, quando o usuário solicitou um novo cadastro, então o sistema envia o protocolo de erro para o e-mail de suporte conforme o protótipo em anexo.

Dado que o sistema verifica que não foi enviado o e-mail de confirmação, quando o usuário solicitou um novo cadastro, então o sistema tenta reenviar o e-mail diariamente na hora configurada (00:00:00) até que o e-mail seja enviado.

Dado que o sistema verifica que o e-mail foi enviado, quando houve falha no ato de cadastro, então o sistema envia uma notificação para o administrador que criou o novo usuário com a mensagem: "O e-mail de confirmação do usuário < nome do usuário > foi enviado na data dd/mm/aaaa às 00:00:00."

- Visualizar -

Dado que o usuário está na lista de usuários, quando acionar a opção visualizar, então o sistema apresenta a tela com os dados do usuário desabilitado para edição.

Dado que o usuário está na tela de visualização, quando acionar a opção "voltar", então o sistema retorna para a tela anteriormente executada.

- Alteração -

Dado que o usuário está na lista de usuário, quando acionar alterar um usuário, então o sistema apresenta o formulário de alteração e realiza as mesmas validações do ato de inserção.

Dado que o usuário está alterando um cadastro, quando acionar a opção cancelar, então o sistema apresenta a mensagem de confirmação com as opções "sim" e "não": "Deseja cancelar a alteração? Esta ação irá descartar as alterações efetuadas no cadastro do usuário."

Dado que o usuário acionou cancelar a alteração, quando acionar a opção "sim", então o sistema retorna para a lista de usuários.

Dado que o usuário acionou cancelar a alteração, quando acionar a opção "não", então o sistema permanece no formulário de alteração.

- Inativar -

Dado que o usuário está na lista de usuário, quando acionar inativar um usuário, então o sistema apresenta a mensagem de confirmação com as opções "confirmar" e "cancelar": "Realmente deseja inativar este usuário? Esta ação irá impossibilitar o usuário < nome do usuário > acessar o sistema.

Dado que o usuário inativou um usuário no sistema, quando o sistema inativar o usuário, então enviar o e-mail de confirmação para o usuário inativado conforme o protótipo em anexo (aplicar a mesma regra de validação de envio de e-mail descrito no ato de inserção).

Dado que o sistema apresentou a mensagem de confirmação de inativação, quando acionar "cancelar", então o sistema cancela a inativação e retorna para a lista de usuários.

- Ativar -

Dado que o usuário acionou ativar um usuário, quando este estiver inativo, então o sistema ativa o usuário, envia um e-mail de confirmação (aplicar a mesma regra de validação de envio de e-mail descrito no ato de inserção) e apresenta a mensagem de sucesso: "O usuário < nome do usuário > foi ativado com sucesso"

B2. – EST02. Autenticar Usuário

SDC - SISTEMA DE DEMONSTRAÇÕES CONTÁBEIS

QUADRO DE TAREFAS

HISTÓRIA DE USUÁRIO

0 #2 EST02 - Autenticar Usuário < >

release 1 × sprint 1 × + Adicionar tag

Criado por Daniel Rabelo
09 mar 2017 15:22 

**Eu como usuário do sistema preciso realizar login ou então recuperar minha senha para que possa realizar as funcionalidades do sistema.
**

Teste de Aceitação:

Dado que o usuário possua conta no sistema, quando informar seu e-mail e senha e selecionar login, então o sistema redireciona o usuário para a página inicial do sistema.

Dado que o usuário solicita autenticar no sistema, quando informar e-mail ou senha incorretos, então o sistema apresenta a mensagem de erro: E-mail ou senha incorretos!

Dado que o usuário solicita autenticar no sistema, quando informar e-mail inexistentes na base de dados, então o sistema apresenta a mensagem de erro: Usuário inexistente, tente outro!

Dado que o usuário solicite alterar a sua senha, quando informar seu e-mail e selecionar recuperar, então o sistema envia para o e-mail do usuário a sua nova senha de acesso e apresenta a mensagem de sucesso: Uma nova senha foi enviado para o e-mail: < e-mail do usuário >.

Dado que o usuário solicite alterar a sua senha, quando informar um e-mail inexistente na base de dados, então o sistema apresenta a mensagem de erro: E-mail inexistente, tente outro.

Dado que o usuário recupere a sua senha, quando o sistema enviar o e-mail, então o usuário poderá visualizar no e-mail as seguintes informações: nome do usuário, data e hora que foi solicitado a nova senha, nova senha gerada e link para login.

Dado que o usuário recupere a sua senha, quando o sistema enviar o e-mail e o usuário não realizar login nas próximas 48 horas (após o envio do e-mail), então o sistema descarta a nova senha gerada e mantém a antiga.

Dado que o usuário recupere a sua senha, quando o usuário realizar login pela primeira vez com a nova senha, então o sistema apresenta a tela de alteração de senha onde o usuário deverá informar a nova senha e confirmar ela.

Dado que o usuário acessou a primeira vez após a recuperação de senha, quando o usuário informar a nova senha, então o sistema apresenta a mensagem de sucesso: Senha alterada com sucesso.

B3. – EST03. Gerenciar Empresa

SDC - SISTEMA DE DEMONSTRAÇÕES CONTÁBEIS

QUADRO DE TAREFAS

HISTÓRIA DE USUÁRIO

▲
0
#3 EST03 - Gerenciar Empresa
<

release 1 ✕
sprint 1 ✕
+ Adicionar tag

Criado por Daniel Rabelo
23 mar 2017 13:36

Eu como administrador do sistema preciso gerenciar os dados da minha empresa, para que possa realizar as demais funções do sistema.

Teste de Aceitação:

Dado que o usuário possui permissão de Administrador, quando acionar a opção "empresa", então poderá visualizar ou alterar os dados da sua empresa.

Dado que o usuário está na tela de visualização de dados da empresa, quando acionar "alterar", então os campos que antes eram desabilitados para edição, agora se tornam habilitados.

Dado que o usuário acionou a opção de alteração, quando visualizar o formulário de alteração, então poderá alterar as informações dos campos:

- CNPJ: Campo obrigatório, de 14 caracteres, com máscara no formato xx.xxx.xxx/xxxx-xx.
- Razão Social: Campo obrigatório, de 150 caracteres.
- CEP: Campo obrigatório, de 8 caracteres, com máscara no formato xx.xxx-xxx.
- UF: Combo obrigatório, previamente selecionado a partir do CEP informado.
- Município: Combo obrigatória, previamente selecionado a partir do CEP informado.
- Bairro: Campo obrigatório, de 150 caracteres. Deverá vir previamente preenchido a partir do CEP informado, caso o CEP possua um bairro associado à ele.
- Complemento: Campo opcional, de 150 caracteres.
- Inscrição Estadual: Campo obrigatório, possui máscara de acordo com a UF selecionada. Verificar anexo as máscaras de cada estado.
- Inscrição Municipal: Campo obrigatório, de 30 caracteres.
- Composição Societária: (Verificar com o Paulo).
- Opção Tributária para o exercício: (Verificar com o Paulo).
- Tipos de serviços e produtos vendidos: Campo opcional, de 250 caracteres.
- Descrição da Empresa: Campo obrigatório, de 250 caracteres.
- Grupo de Empresas: (Verificar com o Paulo)
- Telefone: Campo opcional, de 10 caracteres, possui máscara no formato (xx) xxxx-xxxx.

Dado que o usuário preencheu o formulário de alteração, quando deixar algum campo obrigatório em branco e acionar "salvar", então o sistema retorna a mensagem de impedimento: O(s) seguinte(s) campo(s) obrigatório(s) não pode(m) ficar vazios: <campo 1>, <campo 2>.

Dado que o usuário preencheu o formulário de alteração, quando preencher corretamente e acionar "salvar", então o sistema altera os dados da empresa e retorna a mensagem de sucesso: Alteração efetuada com sucesso!.

Dado que o usuário informa um CEP inválido ou inexistente, quando sair do foco do campo, então o sistema retorna a mensagem de impedimento: CEP informado é inválido ou inexistente, tente outro!

Dado que o usuário informou um CEP válido, quando sair do foco do campo, então o sistema preenche os campos UF, Município e Bairro (este ultimo, apenas se o CEP for referente à um bairro).

Dado que o usuário informou um CNPJ inválido ou inexistente, quando sair do foco do campo, então o sistema retorna a mensagem de impedimento: O CNPJ informado é inválido ou inexistente, tente outro!.

Dado que o usuário informou um CNPJ válido, quando sair do foco do campo, então o sistema preenche o campo Razão Social.

B4. – EST04. Manter Estrutura Operacional

SDC - SISTEMA DE DEMONSTRAÇÕES CONTÁBEIS

QUADRO DE TAREFAS

HISTÓRIA DE USUÁRIO

▲
0 #13 EST04 - Manter Estrutura Operacional < >

+ Adicionar tag

Criado por Daniel Rabelo
23 mar 2017 14:32 

Eu como administrador do sistema preciso inserir, alterar, visualizar ou excluir a estrutura operacional da empresa, para que possa configurar as máscaras multiníveis utilizadas nos planos de contas.

Teste de Aceitação

Dado que o usuário acionou a opção "Configuração", quando o sistema abrir os submenus da opção "Configuração", então o usuário poderá acionar a opção "máscaras".

Dado que o usuário acionou a opção "máscaras", quando o sistema abrir a tela de máscaras multiníveis, então o usuário poderá visualizar a configuração da estrutura operacional das máscaras multiníveis do sistema.

Dado que o usuário está na tela de visualização de máscaras multiníveis, quando acionar a opção "alterar" então o sistema disponibiliza o formulário de alteração.

Dado que o usuário está na tela de alteração de máscaras multiníveis, quando efetuar alguma alteração e acionar "salvar", então o sistema apresenta a mensagem: "A estrutura operacional de máscaras multiníveis foi alterada com sucesso!".

Dado que o usuário está alterando as máscaras multiníveis, quando desejar inserir um novo nível na máscara, então o usuário selecionará o botão "Novo Nível".

RN01: Para cada coluna de nível, o sistema deverá apresentar um identificador de nível sequencial: Ex.: 1 para o nível 1, 2 para o nível 2, 3 para o nível 3 e assim por diante.

RN02: Os campos de nível deverá aceitar apenas valores numéricos. Caso informe um valor que não seja numérico, o sistema deverá apagar o dado inserido.

Dado que o usuário está alterando as máscaras multiníveis, quando desejar inserir um novo "Nome do Padrão", então o usuário selecionará o botão "Novo Padrão".

Dado que o usuário acionou a opção de inserção de nome do padrão, quando o sistema inserir uma nova linha na tabela, então o usuário poderá atualizar o novo nome do padrão.

Dado que o usuário configurou a máscara, quando sair do foco do campo, então o sistema atualiza a coluna máscara conforme a configuração.

RN03: O sistema deverá considerar o padrão de apresentação no modelo: "0.0.000.000", onde "0" é a quantidade de caracteres permitidas para cada nível da máscara. Exemplo:

Nível 1 = 1

Nível 2 = 2

Nível 3 = 5

Então a máscara ficará no formato: 0.00.00000

Dado que o usuário inseriu um novo Nome do Padrão, quando não informar dados nos níveis do novo nome e acionar "salvar", então o sistema apresenta a mensagem de erro: Favor informar a quantidade de caracteres de pelo menos 1 (um) nível.

Dado que o usuário inseriu um novo Nome do Padrão, quando não informar a quantidade de dados do nível 1 e acionar salvar, então o sistema apresenta a mensagem de erro: A quantidade de caracteres do nível 1 é obrigatório.

Dado que o usuário está alterando as máscaras multiníveis, quando desejar "excluir" um Nome do Padrão que não possua dados nos campos, então o sistema apaga a linha referente ao nome do padrão.

Dado que o usuário está alterando as máscaras multiníveis, quando desejar "excluir" um nome do padrão que possua dados nos campos de nível, então o sistema apresenta a mensagem de confirmação com as opções de "sim" e "não": Este nome do padrão possui dados cadastrados nos níveis, deseja realmente efetuar a exclusão, esta ação será irreversível?

Dado que o usuário está solicitando excluir um nome do padrão, quando este nome do padrão estiver sendo utilizado como máscara de alguma conta inserida no plano de contas, então o sistema impede a exclusão e apresenta a mensagem: Este nome do padrão não pode ser excluído pois está sendo utilizado no plano de contas. Altere o plano de contas para poder apagar este nome do padrão.

B5. – EST05. Gerenciar Plano de Contas

SDC - SISTEMA DE DEMONSTRAÇÕES CONTÁBEIS

QUADRO DE TAREFAS

HISTÓRIA DE USUÁRIO

▲
0

#41 EST05 - Gerenciar Plano de Contas

<

+ Adicionar tag

Criado por Daniel Rabelo
24 jul 2017 15:37 

Eu como administrador do sistema preciso importar o plano de contas da minha empresa para o sistema, para que possa alimentar a base de dados do sistema, afim de emitir futuros relatórios.

Teste de Aceitação:

Dado que o usuário acionou a opção "Plano de Contas", quando o sistema carregar a tela de gerenciamento de plano de contas, então o usuário poderá importar ou alterar o plano de contas da empresa.

Dado que o usuário acionou a opção "selecionar arquivo", quando o usuário selecionar documentos do tipo .xls ou .txt, então poderá importar o plano de contas para a base de dados do SDC.

Dado que o usuário selecionou um arquivo que não seja .xls ou .txt, quando o sistema validar a extensão do arquivo, então irá impedir a importação e apresenta a mensagem de erro: "Importação negada! Favor selecionar um arquivo do tipo ".xls" ou ".txt"!".

Dado que o usuário selecionou um arquivo para importação, quando o o CNPJ presente no arquivo for igual ao CNPJ cadastrado na funcionalidade "EST03 - Gerenciar Empresa", então o sistema realiza a importação.

Dado que o usuário selecionou um arquivo para importação, quando o o CNPJ presente no arquivo for diferente ao CNPJ cadastrado na funcionalidade "EST03 - Gerenciar Empresa", então o sistema impede a importação e apresenta a mensagem de erro: "Importação negada! O plano de contas importado deve ser referente à empresa cadastrada no sistema.".

Dado que o usuário está realizando a importação do arquivo, quando o sistema verificar que máscara multinível configurada no sistema (EST04 - Manter Estrutura Operacional) está de acordo com o arquivo importado, então o sistema permite a importação do plano de contas.

Dado que o usuário está realizando a importação do arquivo, quando o sistema verificar que máscara multinível configurada no sistema (EST04 - Manter Estrutura Operacional) não está de acordo com o arquivo importado, então o sistema apresenta a mensagem de erro: "Importação negada! Favor configurar as máscaras multiníveis conforme o plano de contas.".

Dado que o usuário está realizando a importação do arquivo, quando o sistema verificar que máscara multinível configurada no sistema está de acordo com o arquivo importado, então o sistema permite a importação do plano de contas.

Dado que o usuário está realizando a importação do arquivo, quando o sistema finalizar a importação, então o sistema apresenta a mensagem de sucesso: "Importação do plano de contas foi realizada com sucesso!".

Dado que o sistema realizou a importação dos dados do plano de contas para a base de dados do SDC, quando carregar os dados do plano de contas no formulário de importação, então o usuário poderá alterar os dados do plano de contas.

Os dados apresentados na tela são: Código da Conta, Descrição da Conta, Movimento da Conta, Nível da Conta, Grau da Conta e Tipo da Conta.

O usuário poderá editar apenas a descrição da conta, para as contas importadas via arquivo.

Dado que o sistema apresentou as contas na tela, quando o usuário solicitar inserir uma nova conta através da opção "Incluir nova Conta", então o sistema apresenta a tela de inclusão.

Campos do formulário de inclusão (todos campos são obrigatórios):

Código da Conta: Código numérico, deve seguir a máscara configurada no sistema.

Descrição da Conta: Campo alfanumérico, 300 caracteres.

Movimento da Conta: Select com as opções: Seleccione, Crédito e Débito.

Nível: (ver com o paulo)

Grau: (ver com o paulo)

Tipo de Conta: (ver com o paulo)

Dado que o usuário preencheu o formulário de inserção de campo, quando não informar um campo obrigatório e acionar "inserir", então o sistema apresenta a mensagem de erro: "Os seguintes campos obrigatórios estão em branco: < campo1 >, < campo2 > e < campo3 >".

Dado que o usuário está inserindo uma nova conta, quando acionar "inserir", então o sistema insere a conta no plano de contas e apresenta a mensagem de confirmação: "A conta informada foi inserida ao plano de contas com sucesso!".

Dado que o usuário solicitou "alterar", quando o sistema apresentar a tela de alteração de conta, então o usuário poderá alterar apenas a descrição da conta.

Dado que o usuário efetuou uma alteração, quando acionar "alterar", então o sistema apresenta a mensagem de sucesso: "A conta foi alterada com sucesso!".

Dado que o usuário preencheu o formulário de alteração de campo, quando não informar um campo obrigatório e acionar "alterar", então o sistema apresenta a mensagem de erro: "Os seguintes campos obrigatórios estão em branco: < campo1 >, < campo2 > e < campo3 >".

Dado que o usuário solicitou excluir uma conta presente no plano de contas, quando nenhuma outra conta depender da conta selecionada, então o sistema apresenta a mensagem de confirmação com as opções "sim" e "não" "Deseja realmente excluir esta conta? Esta ação é irreversível."

Dado que o usuário solicitou excluir uma conta presente no plano de contas, quando outras contas depender da conta selecionada, então o sistema impede a exclusão e apresenta a mensagem de erro: "As contas <codigo 1>, <codigo 2>, dependem desta conta! Favor remover as contas mencionadas antes de excluir esta conta."


B6. – EST06. Gerenciar Plano de Contas

SDC - SISTEMA DE DEMONSTRAÇÕES CONTÁBEIS

QUADRO DE TAREFAS

HISTÓRIA DE USUÁRIO

0
#68 [EST06] - Gerenciar Balancete
<

release 1 ✕ | sprint 3 ✕ | + Adicionar tag
Criado por Daniel Rabelo  04 set 2017 20:10

Eu como administrador do sistema preciso importar o balancete da minha empresa para o sistema, para que possa alimentar a base de dados do sistema, afim de emitir futuros relatórios.

Teste de Aceitação:

Dado que o usuário acionou a opção "Balancete", quando o sistema carregar a tela de gerenciamento de balancete, então o usuário poderá importar o balancete da empresa.

Dado que o usuário acionou a opção "selecionar arquivo", quando o usuário selecionar documentos do tipo .xls ou .txt, então poderá importar o balancete para a base de dados não relacional do SDC.

Dado que o usuário selecionou um arquivo que não seja .xls ou .txt, quando o sistema validar a extensão do arquivo, então irá impedir a importação e apresenta a mensagem de erro: "Importação negada! Favor selecionar um arquivo do tipo ".xls" ou ".txt"!".

Dado que o usuário está realizando a importação do arquivo, quando o sistema finalizar a importação, então o sistema apresenta a mensagem de sucesso: "Importação do balancete foi realizada com sucesso!".

RN01.: O balancete importado deverá possuir vinculo com o plano de contas ativo no sistema.

RN02.: A opção de menu "Balancete" não poderá ser acessível caso não exista plano de contas ativo no sistema (a validação deverá ser feita através da sessão do usuário).

Os dados apresentados na tela são:

Edit

Dado que o sistema está realizando a importação do balancete, quando verificar que a conta listada no balancete está de acordo com o plano de contas, então o sistema efetua a importação com sucesso.

Dado que o sistema está realizando a importação do balancete, quando verificar que a conta listada no balancete não existe no plano de contas, então o sistema não efetua a importação e apresenta a mensagem de erro: "O balancete não foi importado. Existe contas no balancete que não fazem parte do plano de contas.".

PENDÊNCIA: Analisar a melhor forma de modelar o banco de dados não relacional, para que o sistema tenha um melhor desempenho.

B7. – EST07. Emitir Relatórios de Demonstrações Contábeis

SDC - SISTEMA DE DEMONSTRAÇÕES CONTÁBEIS HISTÓRIA DE USUÁRIO

▲
0 #77 [EST07] - Emitir Relatórios de Demonstrações Contábeis <

+ Adicionar tag Criado por Daniel Rabelo 
15 nov 2017 23:36

Eu como administrador do sistema preciso emitir os relatórios de demonstrações contábeis, para que a tenha um controle do patrimônio da minha empresa, afim de prestar contas para receita.

Teste de Aceitação:

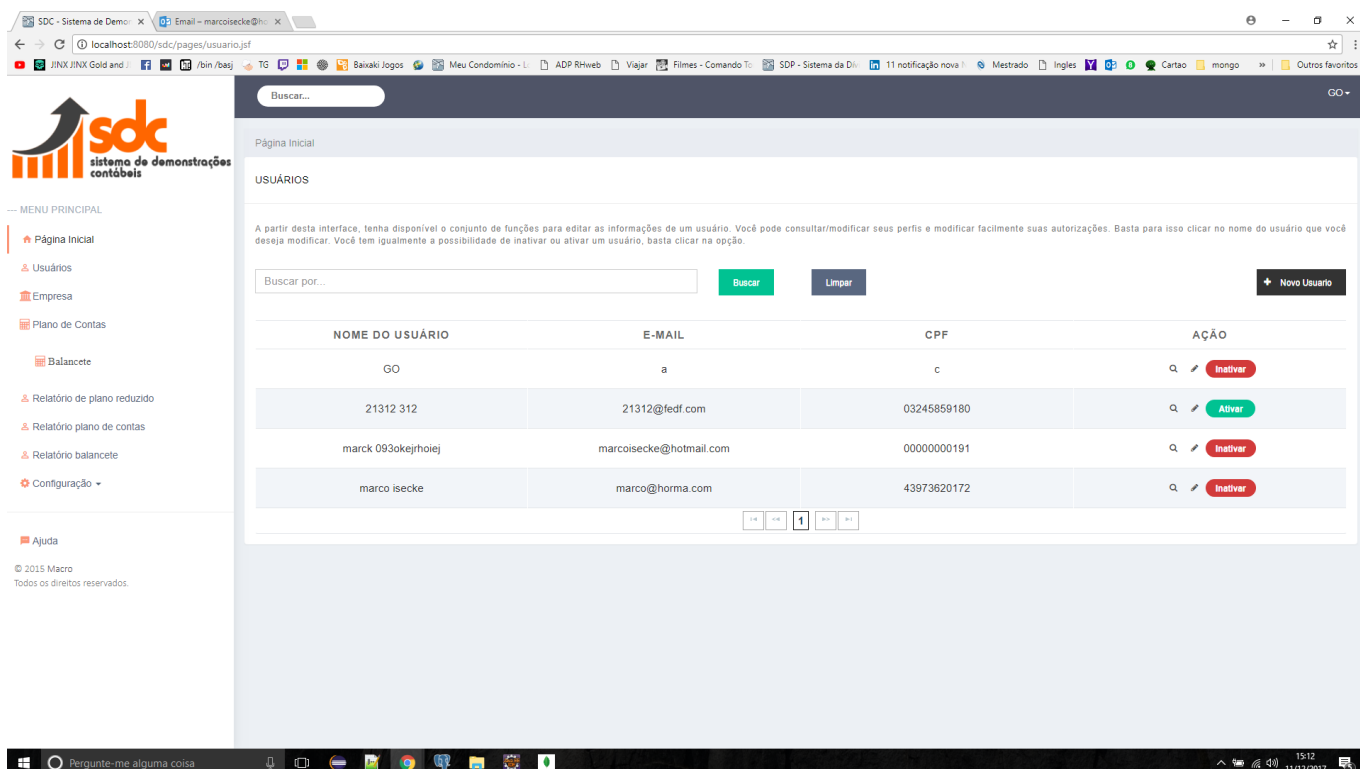
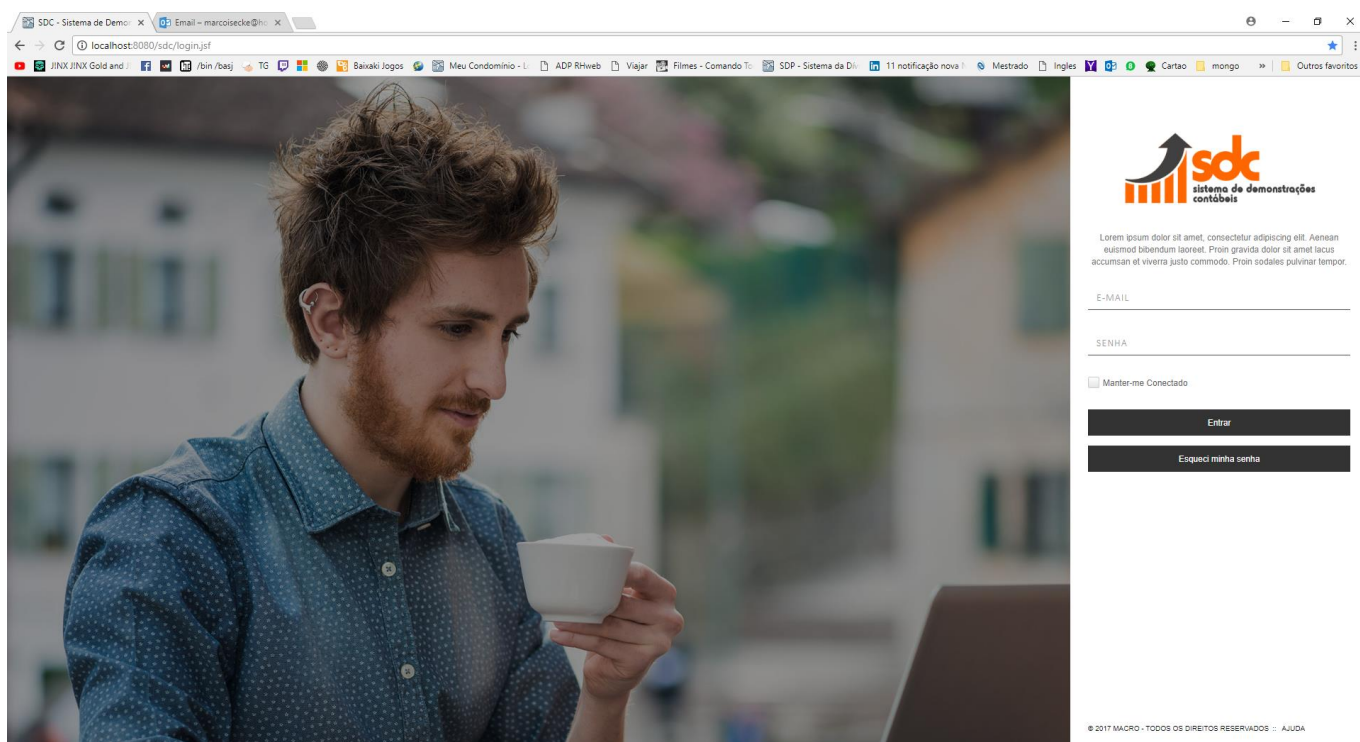
Dado que o usuário acionou a opção "Relatórios" a partir do menu principal, quando o sistema carregar a tela de seleção de relatórios, então o usuário poderá acionar as opções "Balanco Patrimonial", "Demonstrações de Fluxo de Caixa" e "Mutações do Patrimônio Líquido".

Dado que o usuário acionou qualquer uma das opções o sistema irá gerar e disponibilizar para download o relatório selecionado conforme modelo repassado pelo cliente.

RN01. A opção de relatório deve estar desabilitada para seleção caso o sistema não possua Balancete e Plano de Contas importado.

Dado que o usuário emitiu um relatório, então o sistema deve apresentar a mensagem de sucesso "relatório <nome do relatório> emitido com sucesso".

ANEXO D – TELAS DO SISTEMA



SDC - Sistema de Demonstrações Contábeis

localhost:8080/sdc/pages/usuario.jsf

Buscar...

Página Inicial

NOVO USUÁRIO

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusco non dui mattis, molestie diam sit amet, aliquet tortor. Quisque sit amet tincidunt tellus. Pellentesque nec pharetra tortor. In hac habitasse platea dictumst.

Nome *

Sobrenome *

E-mail * Telefone *

Gênero * Estado Civil *

RG * CPF *

Data de Nascimento * CEP *

Estado * Cidade * Bairro * Número

Endereço * Complemento

Pergunte-me alguma coisa

15:13 11/12/2017

SDC - Sistema de Demonstrações Contábeis

localhost:8080/sdc/pages/usuario.jsf

Buscar...

Página Inicial

ALTERAR ADMINISTRADOR

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusco non dui mattis, molestie diam sit amet, aliquet tortor. Quisque sit amet tincidunt tellus. Pellentesque nec pharetra tortor. In hac habitasse platea dictumst.

Nome *

Sobrenome *

E-mail * Telefone *

Gênero * Estado Civil *

RG * CPF *

Data de Nascimento * CEP *

Estado * Cidade * Bairro * Número

Endereço * Complemento

Pergunte-me alguma coisa

15:13 11/12/2017

SDC - Sistema de Demonstrações Contábeis

localhost:8080/sdc/pages/usuario.jsf

Buscar...

Página Inicial

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce non dui mattis, molestie diam sit amet, aliquet tortor. Quisque sit amet tincidunt tellus. Pellentesque nec pharetra tortor. In hac habitasse platea dictumst.

Nome * 21312 Sobrenome * 312

E-mail * 21312@fedf.com Telefone * (87) 2389-17382 Gênero * Masculino Estado Civil * Solteiro(a)

RG * 3242323423 CPF * 032.458.591-80 Data de Nascimento * 02/04/1992 CEP * 75080-050

Estado * GO Cidade * Anápolis Bairro * Maracananzinho Número 3242

Endereço * Rua Augusto de Lima Complemento

Voltar

Pergunte-me alguma coisa

15:14 11/12/2017

SDC - Sistema de Demonstrações Contábeis

localhost:8080/sdc/pages/mascara.jsf

Buscar...

Página Inicial

MÁSCARAS MULTINÍVEIS

As máscaras multiníveis são essenciais para o funcionamento correto do sistema. Estes campos serão consultados durante o processo de importação do plano de contas e balancete, mantenha-os configurados conforme o arquivo que será importado.

Novo Padrão Novo Nível Excluir Padrão

#	NOME DO PADRÃO	1	2	3	4	5	6	7	MÁSCARA
<input type="checkbox"/>	a	1	1	2	2	3	3	0	9.9.99.99.999.999
<input type="checkbox"/>									

Salvar Alterações Cancelar

Pergunte-me alguma coisa

15:15 11/12/2017

SDC - Sistema de Demonstrações Contábeis

Buscar...

Página Inicial

GERENCIAR PLANO DE CONTAS

Macro
67.398.122/0001-59

Visualizar Plano de Contas
Selecione

O Plano de Contas é um arquivo que contém um conjunto de contas, previamente estabelecido, que norteia os trabalhos contábeis. Através desta funcionalidade, o usuário conseguirá importar e gerenciar o plano de contas da sua empresa, para um período de tempo definido.

Período de validade: à [Selecionar Plano de Contas](#)

CÓDIGO DA CONTA	DESCRIÇÃO DA CONTA	NÍVEL	GRAU	TIPO DE CONTA
1	ATIVO			
1.1	ATIVO CIRCULANTE			
1.1.10	DISPONIBILIDADES			
1.1.10.10	CAIXA			
1.1.10.10.010	CAIXA GERAL			
1.1.10.10.010.001	Caixa			
1.1.10.10.012	CAIXA - FUNDO FIXO			
1.1.10.10.012.001	Fundo Fixo			
1.1.10.15	BANCOS			

Perunte-me alguma coisa

15:16
11/12/2017

SDC - Sistema de Demonstrações Contábeis

Buscar...

Página Inicial

BALANCETE

[Selecionar arquivo](#)

CÓDIGO DA CONTA	DESCRIÇÃO DA CONTA	SALDO ANTERIOR	SALDO DÉBITO	SALDO CRÉDITO	SALDO ANTERIOR	SALDO ATUAL
No records found.						

Perunte-me alguma coisa

15:20
11/12/2017

