

**UNIVERSIDADE DE CAXIAS DO SUL
CENTRO DE CIÊNCIAS EXATAS E DAS TECNOLOGIAS
BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

CAMILA RACHEL TONIN DE ANDRADE

INTERGENICDB 2.0

TRABALHO DE CONCLUSÃO DE CURSO

**CAXIAS DO SUL
2016**

CAMILA RACHEL TONIN DE ANDRADE

INTERGENICDB 2.0

Trabalho de Conclusão de Curso do Título de Bacharel pela Universidade de Caxias do Sul. Área de concentração: Sistemas de Informação.

Orientador: Prof. Dr. Daniel Luis Notari

CAXIAS DO SUL

2016

Aos meus pais que sempre me apoiaram e incentivaram nesta jornada.
Ao meu amado esposo por sua paciência, dedicação, apoio e por sempre acreditar nas minhas competências.

AGRADECIMENTOS

Agradeço a meu esposo, Eduardo, pelo companheirismo, dedicação, renúncia e compreensão durante todos esses anos de estudo. Aos meus pais, Ivadir e Idi, pela formação e apoio incondicional que me acompanham até hoje.

Ao meu orientador, Prof. Dr. Daniel Luis Notari pela orientação segura, conhecimento, disponibilidade, paciência e sugestões acertadas que tão bem me conduziram para alcançar este objetivo.

Por fim, agradeço a todos os professores, grandes mestres, que transmitiram seus conhecimentos, suas experiências, dedicaram seu tempo durante esses anos. Este trabalho não seria possível sem o conhecimento adquirido de cada um no decorrer desta jornada.

RESUMO

O portal IntergenicDB é um repositório público no qual pesquisadores podem consultar as informações referentes a dados integrados de regiões intergênicas com as informações dos seus genes. Na versão 1.0 o portal IntergenicDB utiliza o SGBD MySQL para gerenciar os seus dados. O MySQL possui algumas limitações que estão dificultando a utilização dos usuários administrativos do portal. Este usuário enfrenta problemas de lentidão no acesso ao sistema; e dificuldade na criação de *scripts* SQL compatíveis com o MySQL. O *site* de hospedagem utilizado não fornece as versões mais atuais do MySQL. Isto gera erros nas consultas executadas e nas exportações dos dados. Por estes motivos, optou-se por migrar a base de dados existente para o SGBD PostgreSQL. Para realizar a migração foi realizado um estudo dos métodos de migração existentes, com o objetivo de se definir a forma mais adequada para transferir os dados da parte administrativa do portal. Durante a realização da migração, verificou-se problemas nos dados existentes referentes a biologia molecular. Por isso, foram realizadas adequações nas estruturas das tabelas existentes, uma nova carga de dados foi realizada e, vários testes para garantir a integridade dos dados foram realizados. As estruturas e dados das tabelas foram migradas com sucesso. Foi identificado que o problema de desempenho das consultas possuía ligação com o *framework* LP.Framework utilizado para acessar os dados no MySQL. A retirada do LP.Framework proporcionou um novo rumo a proposta inicial. O portal foi migrado para plataforma de desenvolvimento PHP. Esta alteração proporcionou uma evolução no *layout* do portal e a inclusão de novas funcionalidades.

Palavras-chave: IntergenicDB. Banco de dados. Migração de dados.

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama simplificado da arquitetura do sistema de banco de dados.....	12
Figura 2 – Ranking dos SGBDs do DB-Engines.....	13
Figura 3 – Distribuição do esforço de manutenção.....	19
Figura 4 – Um modelo de processo de reengenharia de software.....	21
Figura 5 – Processo de ETL através de bases de dados semelhantes.....	24
Figura 6 – Exemplo de arquitetura da migração <i>Chicken Little</i>	25
Figura 7 – Exemplo de arquitetura da migração <i>Butterfly</i>	27
Figura 8 – Comparação entre <i>script</i> MySQL e PostgreSQL.....	34
Figura 9 – Arquitetura orientada a serviços.....	36
Figura 12 – Tela de listagem do novo cadastro de ajuda.....	42
Figura 13 – Tela do novo cadastro de categoria.....	42
Figura 14 – Tela do software TortoiseGit.....	44
Figura 15 – Arquitetura do portal IntergenicDB.....	47
Figura 16 – Novo <i>layout</i> do Portal IntergenicDB.....	48
Figura 17 – Novo <i>layout</i> da tela de <i>login</i> da área administrativa.....	49
Figura 18 – Tela inicial do portal administrativo do IntergenicDB 2.0.....	49
Figura 19 – Área do <i>site</i> para acessar o link para recuperar a senha.....	53
Figura 20 – Novas funcionalidades para o usuário logado.....	55
Figura 21 – Permissões do cadastro de grupos.....	56
Figura 22 – Tela de cadastro de usuário na área administrativa.....	57
Figura 23 – Modelo ER da área administrativa do portal IntergenicDB.....	60
Figura 24 – Tempo de carregamento da página de consulta de Gene.....	61
Figura 25 – Tempo de carregamento da página de consulta de Gene/Organismo....	61
Figura 26 – Tempo de carregamento da página de consulta de Gene no IntergenicDB 2.0.....	61
Figura 27 – Tempo de carregamento da página de consulta de Gene/Organismo no IntergenicDB 2.0.....	61
Figura 28 – <i>E-mail</i> que o usuário receberá ao recuperar a senha.....	62
Figura 29 – Tela de cadastro do texto ajuda.....	65
Figura 30 – Cadastro de categorias na área administrativa.....	66
Figura 31 – Tela de cadastro do texto da página inicial do <i>site</i>	67
Figura 32 – Texto da página inicial na área do <i>site</i>	67

Figura 33 – Cadastro de publicação na área administrativa do portal.....	68
Figura 34 – Tela de publicações na área do <i>site</i>	68
Figura 35 – Exemplo de <i>e-mail</i> enviado para um novo contato realizado pelo <i>site</i> ...	69
Figura 36 – <i>E-mail</i> recebido ao ter enviado um arquivo pelo site.....	70
Figura 37 – Tela de consulta no IntergenicDB 2.0.....	71
Figura 38 – Simulação da visualização do portal em um iPad.....	71
Figura 39 – Comparação do <i>layout</i> iPhone 6 x Samsung Galaxy S5.....	72
Figura 40 – Página inicial do portal IntergenicDB.....	80
Figura 41 – Página inicial da área administrativa.....	81
Figura 42 – Arquitetura do portal IntergenicDB.....	82

LISTA DE TABELAS

Tabela 1 – Equivalência de tipos de dados entre SGBDs.....	22
Tabela 2 – Comparativo entre as metodologias ETL, <i>Chicken Little</i> e <i>Butterfly</i>	28
Tabela 3 – Relacionamento entre verbos SQL e HTTP.....	37
Tabela 4 – Operações disponibilizadas pelo <i>web service</i>	38
Tabela 5 – Parâmetros do método utilizando protocolo POST.....	38
Tabela 6 – Parâmetros do método utilizando protocolo PUT.....	39
Tabela 7 – Parâmetros do método utilizando protocolo GET.....	39
Tabela 8 – Tabelas e quantidade de registros que foram importados para o PostgreSQL.....	46
Tabela 9 – <i>Status</i> HTTP que podem retornar do <i>web service</i>	50
Tabela 10 – Parâmetros e retorno do método <i>users</i> utilizando POST.....	51
Tabela 11 – Parâmetros e retorno do método <i>users</i> utilizando PUT.....	51
Tabela 12 – Parâmetros e retorno do método <i>login</i> utilizando GET.....	52

LISTA DE ABREVIATURAS E SIGLAS

DDL	<i>Data Definition Language</i>
ERP	<i>Enterprise Resource Plannig</i>
ETL	Extrair, Transformar e Carga
HTML	<i>HyperText Markup Language</i>
IIS	<i>Internet Information Services</i>
MVC	Modelo Visão Controlador
REST	Transferência de Estado Representacional
SCV	Sistema de Controle de Versões
SGBD	Sistema de Gerenciamento de Banco de Dados
SGI	Sistema de Gestão Integrada
SOAP	<i>Simple Object Access Protocol</i>
SQL	<i>Structured Query Language</i>
SVN	<i>Subversion</i>
UCS	Universidade de Caxias do Sul
UDDI	<i>Universal Description, Discovery and Integration</i>
W3C	<i>World Wide Web Consortium</i>
WCF	<i>Windows Communication Foundation</i>
WSDL	<i>Web Services Description Language</i>
XML	<i>Extensible Markup Language</i>

SUMÁRIO

1 INTRODUÇÃO	12
1.1 CONTEXTUALIZAÇÃO.....	12
1.2 PROBLEMA DE PESQUISA.....	14
1.3 QUESTÃO DE PESQUISA.....	15
1.4 OBJETIVOS.....	15
1.4.1 Objetivo Geral	16
1.4.2 Objetivos Específicos	16
1.5 ESTRUTURA DO TEXTO.....	16
2 REFERENCIAL TEÓRICO	18
2.1 EVOLUÇÃO DE SOFTWARE.....	18
2.1.1 Manutenção de Software	18
2.1.2 Reengenharia de software	20
2.2 MIGRAÇÃO DE DADOS.....	22
2.3 METODOLOGIAS DE MIGRAÇÃO DE DADOS.....	23
2.3.1 Extact, Transform e Load (ETL)	23
2.3.2 Chicken Little	24
2.3.3 Butterfly	26
2.3.4 Comparativo	27
2.3.5 Trabalhos Relacionados	28
2.4 FERRAMENTAS DE MIGRAÇÃO DE DADOS.....	29
2.5 CONSIDERAÇÕES FINAIS.....	31
3 PROPOSTA DE SOLUÇÃO	32
3.1 MIGRAÇÃO DE DADOS.....	32
3.1.1 Extração	33
3.1.2 Transformação	33
3.1.3 Carga	34
3.2 REALIZAÇÃO DE TESTE.....	34
3.3 CRIAÇÃO DE WEB SERVICE PARA FORNECER DADOS DE <i>LOGIN</i>	35
3.3.1 SOAP	35
3.3.2 REST	36
3.3.3 Resolução	37
3.4 RECUPERAR SENHA.....	39
3.5 CADASTRO DE AJUDA.....	40

3.6 TROCA DE SISTEMA DE CONTROLE DE VERSÕES.....	43
4 IMPLEMENTAÇÃO.....	45
4.1 MIGRAÇÃO DE DADOS.....	45
4.1.1 Migração das tabelas.....	45
4.1.2 Modificação na arquitetura de software.....	47
4.2 CRIAÇÃO DO WEB SERVICE.....	50
4.3 RECUPERAR SENHA.....	52
4.4 CADASTRO DE AJUDA.....	53
4.5 TROCA CONTROLE DE VERSÕES.....	53
4.6 NOVAS FUNCIONALIDADES.....	54
4.6.1 Texto página inicial do site.....	54
4.6.2 Meus dados e Troca de Senha.....	54
4.6.3 Cadastro de Publicações.....	55
4.6.4 Formulário de Contato.....	55
4.6.5 Cadastro de Grupo de Administradores.....	56
4.6.6 Cadastro de Usuários.....	57
4.7 CONSIDERAÇÕES FINAIS.....	58
5 ESTUDO DE CASO.....	59
5.1 MIGRAÇÃO DOS DADOS.....	59
5.2 RECUPERAR SENHA.....	62
5.3 TESTES <i>WEB SERVICE</i>	62
5.4 CADASTRO DE AJUDA.....	65
5.5 NOVAS FUNCIONALIDADES.....	66
5.5.1 Texto página inicial do site.....	66
5.5.2 Cadastro de publicações.....	68
5.5.3 Contato.....	69
5.5.4 Envio de arquivo.....	69
5.6 <i>LAYOUT</i>	70
5.6.1 Área administrativa.....	70
5.6.2 Adaptável as resoluções.....	71
5.7 PUBLICAÇÃO NO SERVIDOR.....	72
5.7.1 Internacionalização.....	73
5.7.2 Arquivo .HTACCESS.....	73
5.8 CONSIDERAÇÕES FINAIS.....	74

6 CONCLUSÃO.....	75
6.1 TRABALHOS FUTUROS.....	76
REFERÊNCIAS.....	77
APÊNDICE A – PORTAL INTERGENIGDB.....	80
ANEXO A – MODELO ER ÁREA ADMINISTRATIVA DO PORTAL.....	83

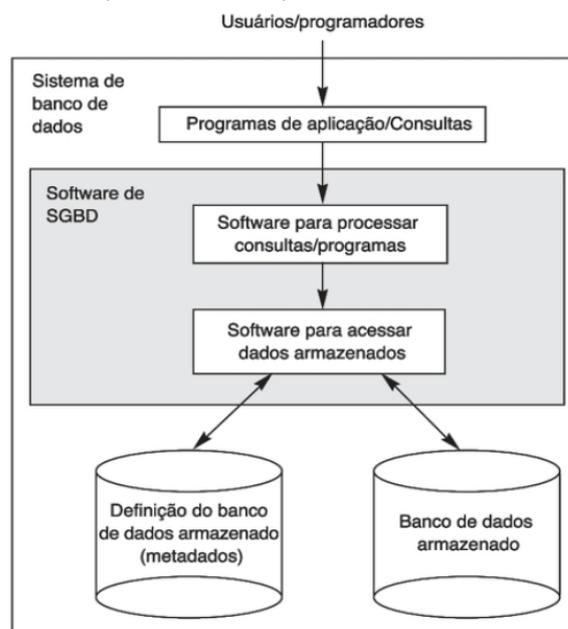
1 INTRODUÇÃO

Neste capítulo será apresentada uma contextualização do trabalho, além da explanação da questão e do problema de pesquisa identificado. Serão destacados os objetivos a serem alcançados durante a execução do mesmo e uma amostragem da estrutura do trabalho durante os próximos capítulos.

1.1 CONTEXTUALIZAÇÃO

Um banco de dados é uma coleção de dados relacionados armazenados que são utilizados pelos sistemas de aplicação para atender uma comunidade de usuários (ELMASRI E NAVATHE, 2011; HEUSER, 2011). Um Sistema de Gerência de Banco de Dados (SGBD) é uma coleção de programas que permite aos usuários criar e manter um banco de dados. O objetivo principal de um SGBD é proporcionar um ambiente eficiente para o armazenamento e para a recuperação das informações no banco de dados (ELMASRI E NAVATHE, 2011).

Figura 1 – Diagrama simplificado da arquitetura do sistema de banco de dados



Fonte: Elmasri e Navathe (2011).

O SGBD (Figura 1) proporciona ao usuário final uma visualização integrada e única dos dados do banco, pois este recebe todas as solicitações dos programas de aplicação/consultas e as traduz nas operações complexas necessárias para atendê-las (ROB E CORONEL, 2011).

Segundo o ranking do DB-Engines¹ (2016), os mais difundidos no mercado são os softwares proprietários Oracle, Microsoft SQL-Server, IBM DB2 e os livres são MySQL, PostgreSQL, MongoDB e SQL-Lite, estes dois últimos em crescente utilização por serem mais adequados a dispositivos móveis, conforme é possível verificar na Figura 2.

Figura 2 – Ranking dos SGBDs do DB-Engines

Rank			DBMS	Database Model	Score		
Jun 2016	May 2016	Jun 2015			Jun 2016	May 2016	Jun 2015
1.	1.	1.	Oracle	Relational DBMS	1449.25	-12.78	-17.11
2.	2.	2.	MySQL 	Relational DBMS	1370.13	-1.69	+91.78
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1165.81	+22.99	+47.76
4.	4.	 5.	MongoDB 	Document store	314.62	-5.60	+35.57
5.	5.	 4.	PostgreSQL	Relational DBMS	306.60	-1.01	+25.70
6.	6.	6.	DB2	Relational DBMS	188.57	+2.61	-10.12
7.	7.	 8.	Cassandra 	Wide column store	131.12	-3.38	+22.21
8.	8.	 7.	Microsoft Access	Relational DBMS	126.22	-5.35	-20.27
9.	 10.	9.	SQLite	Relational DBMS	106.78	-0.48	-1.19
10.	 9.	10.	Redis 	Key-value store	104.49	-3.74	+9.00

Fonte: DB-Engines² (2016)

A migração da informação, de uma parte essencial de um sistema para outro, e de transferência de dados, de um ambiente para outro, estão estreitamente inter-relacionadas com os processos, podendo ser divididas em migração de dados (BROY, 2005). O processo de migração de dados deve ser aplicado e validado com devido cuidado, para evitar a perda de informações ou gerar inconsistências nas mesmas (GUEDES, 2014). A migração de dados significa mover os dados de um

¹ Site de ranking de SGBDs, atualizado mensalmente, classificando-os de acordo com sua popularidade

² Disponível em: <http://db-engines.com/en/ranking/relational+dbms>. Acesso em: 27 mai 2016.

SGBD para outro SGBD e consiste em três passos: extração dos dados, transformação dos dados (se necessário) e carga de dados.

1.2 PROBLEMA DE PESQUISA

O portal IntergenicDB é um repositório de acesso público no qual pesquisadores podem consultar as informações, contidas em seu SGBD, que são referentes a dados integrados de regiões intergênicas com as informações dos seus genes (DALZUCHIO, 2014; NOTARI et al., 2014). Essas informações são inseridas por usuários que possuem um cadastro no sistema e que se utilizam principalmente de importação de dados de outras bases públicas. O portal possui uma ferramenta de administração desses usuários e do portal.

Atualmente o software IntergenicDB está utilizando o SGBD MySQL. O MySQL foi criado na Suécia, com os objetivos de ser estável, seguro, rápido e pouco custoso, devido ao fato de no mercado já existirem outros SGBDs com tais características, porém com custos elevados. É o banco de dados de código aberto mais popular, ocupando o 2º lugar no ranking mundial segundo o DB-Engines (2016) e é considerado rápido, barato, está disponível para a maioria das plataformas de sistemas operacionais e atende sistemas Web e incorporados (MySQL, 2016).

O MySQL demorou um pouco para incrementar funções que já eram comuns em outros SGBD. Por exemplo, *views* só foram incluídas na versão 4.1 e no momento estamos na 5.7, funções como *COMMIT* E *ROLLBACK* não são nativas, devem ser feitas certas configurações para que sejam suportadas (MySQL, 2016). O sistema IntergenicDB pode ter sido criado quando a versão utilizada não disponibilizava certas funções que auxiliariam no bom desempenho do sistema.

O MySQL possui algumas limitações que entram em conflito com as necessidades do IntergenicDB e seus usuários. O *site* de hospedagem no qual o IntergenicDB está disponibiliza uma ferramenta online de administração de bases de dados gratuitamente, o phpMyAdmin. É um *software* que suporta a realização de

diversas operações como: administrar base de dados, tabelas, colunas, relações, índices, usuários, permissões, consulta de dados, importação de dados, exportação de dados, entre outros. O IntergenicDB é um *software* que possui uma grande quantidade de informações armazenadas e o phpMyAdmin não está conseguindo realizar tais operações com êxito e corretamente.

Outra limitação para usuários mais *experts* em SGBD e familiarizados com a Linguagem de Consulta Estruturada, do inglês *Structured Query Language* (SQL), para realizar consultas diretas por meio da ferramenta phpMyAdmin é que o MySQL não segue o padrão SQL e por isso possui comandos que devem ser realizados de forma diferente. Por exemplo, sub selects com a cláusula IN não são suportados até a versão 5.6, isso quer dizer, que uma consulta “select * from table where id in (select id from table2)” não é possível ser realizada, o que é comum em outros SGBDs (SUEHRING, 2002). Isso causa certa dificuldade, já que é necessário pesquisar como executar determinados comandos e consultas, enquanto que em outros SGBD tais comandos seguem o padrão e o usuário já tem tal conhecimento.

1.3 QUESTÃO DE PESQUISA

Diante das necessidades apontadas, a questão de pesquisa a ser respondida é: A migração de dados para o SGBD PostgreSQL é viável, considerando que não ocorram perdas de dados e de um modo que o portal administrativo do IntergenicDB continue funcionando perfeitamente?

1.4 OBJETIVOS

Os objetivos desta proposta dividem-se em objetivo geral e objetivos específicos.

1.4.1 Objetivo Geral

Este trabalho tem como objetivo definir uma estratégia de migração de dados do SGBD MySQL para SGBD PostgreSQL sobre a área administrativa do portal IntergenicDB.

1.4.2 Objetivos Específicos

Para alcançar o objetivo geral, são definidos alguns objetivos específicos, que será realizado ao longo do trabalho de conclusão:

- 1) Migrar os dados do portal administrativo do IntergenicDB do MySQL para o PostgreSQL.
- 2) Criar um *web service* de integração dos usuários do IntergenicDB com outros sistemas contemplando as ações de criar usuário, autenticar (validar usuário e senha) e trocar senha.
- 3) Corrigir *bug* existente nas ações de *login* e recuperar senha do portal de IntergenicDB.
- 4) Desenvolver o cadastro de manutenção da tela de ajuda (*Help*).
- 5) Migrar o código fonte do repositório SVN para GIT.

1.5 ESTRUTURA DO TEXTO

Prevendo alcançar os objetivos acima, este trabalho foi estruturado da seguinte forma: uma conceituação referente aos termos envolvidos na migração de dados do portal administrativo do IntergenicDB e o estudo das metodologias existentes sobre migração bem como ferramentas que podem auxiliar a atender esta tarefa. Após esse estudo, inicia-se a proposta de solução a ser realizada no *software* IntergenicDB. Com base na proposta descrita, é iniciada a migração dos dados e a implementação das alterações para atingir os objetivos propostos. Ao finalizar a

implementação, é criado um estudo de caso e aplicados testes na aplicação para garantir que os objetivos propostos sejam atendidos. Por fim, uma conclusão mostrando os resultados obtidos.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta um referencial teórico sobre os conceitos computacionais utilizados para facilitar o entendimento da proposta da migração de dados do portal IntergenicDB para o SGBD PostgreSQL, a análise de metodologias de migração de dados e ferramentas disponíveis para a realização da migração.

2.1 EVOLUÇÃO DE SOFTWARE

Segundo Sommerville (2011), o processo de desenvolvimento é um processo contínuo por toda vida útil do *software* e não um processo interrompido quando o mesmo é entregue. Pressman (2011) ressalta que a alteração em um sistema é um fato normal, pois clientes necessitam modificar requisitos, desenvolvedores querem modificar a abordagem técnica e gerentes desejam modificar a estratégia do projeto. Qualquer mudança ou defeito pode desencadear uma evolução no *software*, até mesmo uma alteração de *hardware* onde o *software* é utilizado pode acarretar em mudanças para atender a necessidade (PRESSMAN, 2011; SOMMERVILLE, 2011).

2.1.1 Manutenção de Software

A ISO/IEC (1998) 12207 define o ciclo de vida de um *software* por meio de agrupamento de processos que são divididos em quatro classes: fundamentais, apoio, organizacionais e adaptação. O processo de manutenção de *software* está classificado nessa ISO como um processo fundamental para a implementação do *software*.

Segundo Sommerville (2011), existem três tipos de manutenção de *software*:

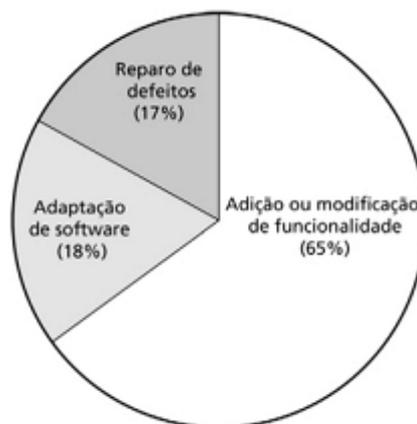
- **Correção de defeitos:** realizada para controlar funções cotidianos do sistema, corrigindo *bugs* de funcionalidades do programa.
- **Adaptação ambiental:** visa a implementação de modificações secundárias

que têm como origem uma modificação em outra parte do sistema ou do meio externo. Por exemplo, adequar o sistema à troca da plataforma do sistema operacional.

- **Adição de funcionalidades:** objetivam realizar acréscimo de novas funcionalidades ou mudanças no *software* independente da existência de defeitos.

Sommerville (2011) ainda ressalta que, segundo pesquisas, a manutenção de *software* ocupa uma proporção maior dos orçamentos de TI que o desenvolvimento, detendo dois terços do orçamento contra um terço do desenvolvimento. As pesquisas também revelam que se gasta mais do orçamento de manutenção na implementação de novos requisitos do que na correção de *bugs* (SOMMERVILLE, 2011). A Figura 3, demonstra essa distribuição de custos de manutenção.

Figura 3 – Distribuição do esforço de manutenção



Fonte: Sommerville (2011).

Percebe-se ainda na Figura 3, que evoluir o sistema para lidar com novos ambientes e novos requisitos consome mais esforço de manutenção. Tais porcentagens podem variar de organização para outra, mas universalmente a correção de falhas não é a atividade de manutenção mais cara (SOMMERVILLE, 2011).

2.1.2 Reengenharia de software

A reengenharia de *software* é uma etapa que pode ser necessária para a realização das mudanças (SOMMERVILLE, 2011) já que algumas mudanças podem gerar uma reformulação do projeto inicial, uma adaptação para uma nova tecnologia, ou apenas refatoramento das funcionalidades do *software* utilizando novas funcionalidades existentes na linguagem utilizada.

A reengenharia pode envolver análise de inventário, reestruturação de documentos, refatoração da arquitetura do sistema, modernização da linguagem de programação, além da reestruturação de dados por meio de atualizações de estruturas de dados do sistema (SOMMERVILLE, 2011; PRESSMAN E MAXIM, 2016).

Segundo Pressman e Maxim (2016), o objetivo das atividades incluídas na reengenharia é criar versões dos programas que apresentam uma qualidade mais alta e melhorar a manutenibilidade. A aplicação da reengenharia proporciona dois benefícios importantes (SOMMERVILLE, 2011): i) risco reduzido, pois desenvolver novamente um *software* crítico de negócios tem um risco muito alto, e, ii) custo reduzido, pois o custo da reengenharia é significativamente menor do que o desenvolvimento de uma nova aplicação.

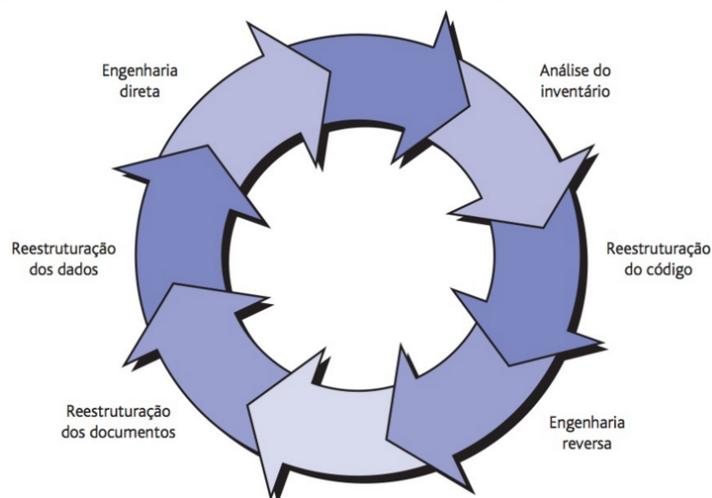
O processo de reengenharia (Figura 4) possui as seguintes atividades (SOMMERVILLE, 2011; PRESSMAN E MAXIM, 2016):

- **Análise do inventário:** Pode ser uma planilha com informações detalhadas como por exemplo, tamanho, idade e criticalidade nos negócios de cada *software* ativo.
- **Reestruturação de documentos:** a documentação deve ser atualizada. A organização pode optar em documentar as partes do *software* que estão atualmente passando por mudanças e com o tempo, documentar o restante.
- **Engenharia reversa:** o *software* é analisado e a partir dele são extraídas

as informações com o intuito de criar uma representação do programa em nível maior de abstração do que o código fonte. Isso auxilia a documentar sua organização e funcionalidade;

- **Reestruturação do código:** utilização de ferramentas de tradução, onde o programa é convertido a partir de uma linguagem de programação antiga para uma versão mais moderna da mesma linguagem ou outra linguagem diferente;
- **Reestruturação de dados:** uma atividade de grande escala onde os objetos e atributos são identificados e a estrutura de dados existente é revisada visando atender as modificações. Isso pode significar alterações no esquema de banco de dados e conversões de dados existentes em novas estruturas, bem como alteração no código-fonte;
- **Engenharia direta:** em um mundo ideal, os aplicativos seriam recriados por meio de um “motor de reengenharia” automatizado.

Figura 4 – Um modelo de processo de reengenharia de software



Fonte: Pressman e Maxim (2016).

Tais atividades podem ser executadas conforme a necessidade de manter a aplicação, não sendo necessário a utilização de todas, pois algumas dessas atividades dependem de como foi realizada a arquitetura do *software*.

2.2 MIGRAÇÃO DE DADOS

A migração de dados significa mover os dados de um SGBD para outro SGBD. Embora existam padrões de tipos de dados no SQL ANSI/ISO, cada SGBD possui os seus objetos de apoio ao gerenciamento de estrutura e dados. Por exemplo, um campo com tipo numérico pode ser classificado NUMERIC, NUMBER, FLOAT, INT, INTEGER, INT8, INT16 etc, dependendo o SGBD (OLIVEIRA E MARCELINO, 2012).

Tabela 1 – Equivalência de tipos de dados entre SGBDs

TIPOS DADOS	SQL ANSI	ORACLE	POSTGRESQL	MYSQL	SQLLITE
INTEIRO	SMALLINT, INTEGER	NUMBER	SMALLINT, INTEGER, BIGINT	TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT	INTEGER
REAL	REAL, DOUBLE, PRECISION, FLOAT	BINARY_FLOAT	REAL, DOUBLE, PRECISION	FLOAT, DOUBLE, REAL	REAL, FLOAT, DOUBLE
DECIMAL	DECIMAL	BINARY_DOUBLE, NUMBER	DECIMAL, NUMERIC	DECIMAL	REAL, FLOAT, DOUBLE
TEXTO	CHARACTER, CHARACTER VARYING, NATIONAL CHARACTER, NATIONAL CHARACTER VARYING	CHAR, VARCHAR2, CLOB, NCLOB, NVARCHAR2, NCHAR	CHAR, VARCHAR, TEXT	CHAR, BINARY, VARCHAR, VARBINARY, TEXT, TINYTEXT, MEDIUMTEXT , LONGTEXT	TEXT, CHAR, CLOB
BINÁRIO	BIT, VARYING	BLOB, RAW, LONGRAW, BFILE	BYTEA	TINYBLOB, BLOB, MEDIUMBLO B, LONGBLOB	NÃO EXISTE
DATA/TEMPO	DATE, TIME, TIMESTAMP	DATE, TIMESTAMP	DATE, TIME, TIMESTAMP	DATETIME, DATE, TIMESTAMP, YEAR	NÃO EXISTE
LÓGICO	NÃO EXISTE	NÃO EXISTE	BOOLEAN	BOOLEAN, BOOL	NÃO EXISTE

Fonte: Adaptado de Oliveira e Marcelino (2012).

Conforme Oliveira e Marcelino (2012) os dados de cada campo de uma tabela podem variar em tamanho ou formato e cada SGBD possui características quanto à precisão, número de bits utilizados, para cada tipo de dados. Por isso após uma análise dos principais tipos de dados dos SGBDs, a Tabela 1 apresenta a comparação entre eles.

2.3 METODOLOGIAS DE MIGRAÇÃO DE DADOS

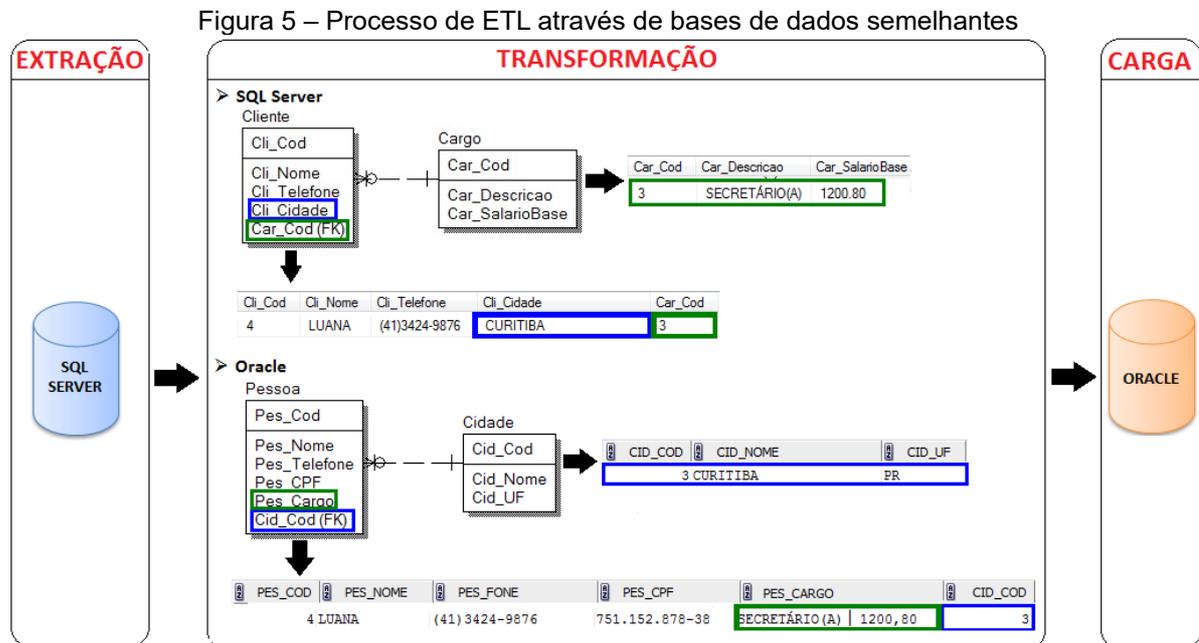
O objetivo de toda migração de *software* é manter dados e funcionalidades em condições de uso no *software* afetado. Para que esse objetivo seja cumprido apresenta-se três metodologias: ETL, *Chicken Little*, *Butterfly*.

2.3.1 Extact, Transform e Load (ETL)

É uma metodologia que se divide em três estágios: Extração, Transformação e Carga, conforme o nome sugere. Primeiro é realizada a extração de dados, processo de extrair os dados de um sistema existente e armazená-los. Segundo, é realizada a transformação dos dados extraídos, se necessário, e também é possível realizar o descarte de dados que não sejam necessários ou melhorias a fim de se obter melhor desempenho. Por fim, é realizada a carga de dados, isto é, transferir o conteúdo do arquivo gerado na etapa de extração para a base de dados da aplicação destino (TRIPATHY E NAIK, 2015).

A Figura 5 mostra a representação das etapas do processo ETL considerando a extração dos dados na base de origem (SQL Server), sua transformação e posteriormente a carga desses dados no SGBD de destino (Oracle). Oliveira e Marcelino (2012), afirmam que esta metodologia é adequada para um ambiente homogêneo, pois é um processo prático de transposição de dados, quando a estrutura de destino é compatível com a origem, criando apenas uma camada para tratamento simples dos dados e logo disponibilizando uma saída

própria para carregamento direto dos dados no sistema de destino.



Fonte: Malacrida et al (2014)

2.3.2 Chicken Little

É uma estratégia de migração que tem como principal característica a migração incremental, onde os sistemas de informação legado³ e destino são operados em paralelo ao longo da migração. No início, o sistema destino é muito pequeno, talvez apenas uma aplicação com um pequeno banco de dados. No entanto, a medida que a migração avança, o sistema destino consequentemente também crescerá até que ele compreenda todas as funcionalidades do sistema legado que poderá ser aposentado (BRODIE, 1993; OLIVEIRA E MARCELINO, 2012).

Durante a migração, o sistema legado e o sistema destino precisam se trocar dados para formar o sistema em produção. Essa troca é realizada através de

³ Segundo Pressman (2011), é um eufemismo para sistemas, geralmente antigos, mal documentados e mal projetados. Para outros autores, qualquer software em produção de uma organização é um sistema legado.

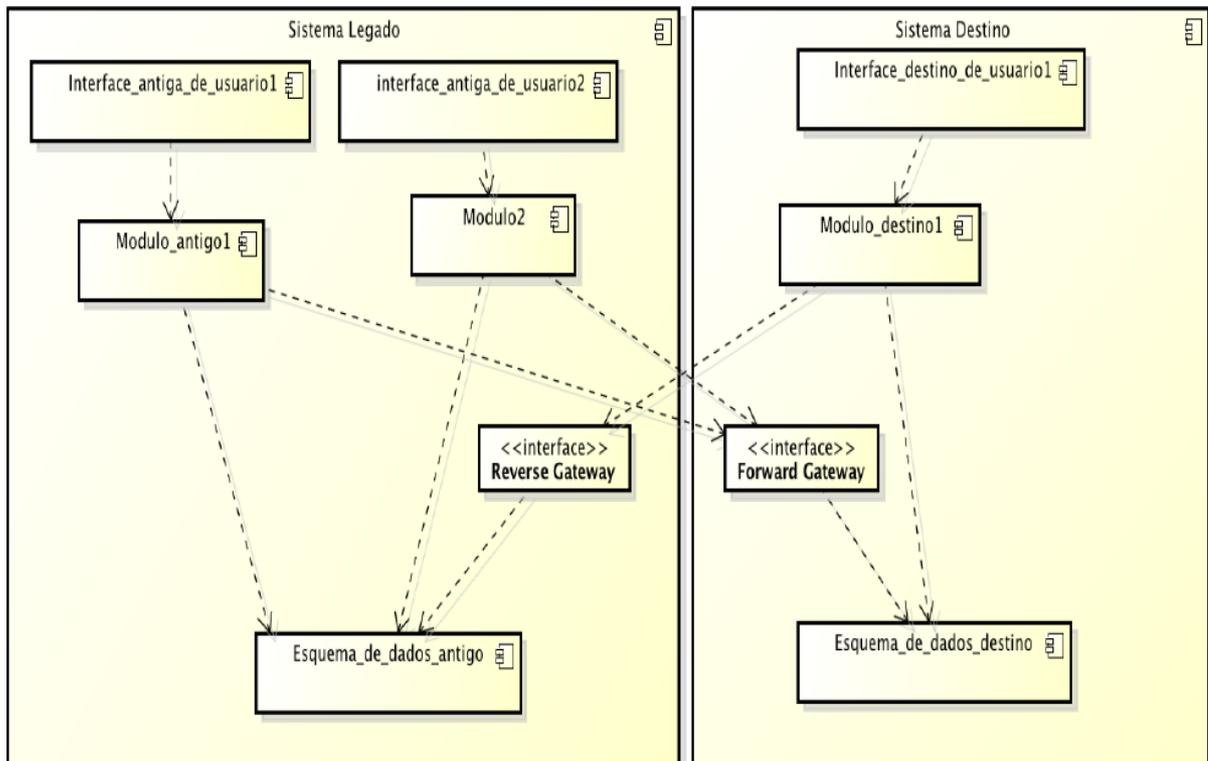
gateways. Para Oliveira e Marcelino (2012), um *gateway* pode ser definido como um programa intermediário responsável pela transformação da informação. Em todo o processo de migração, os dados são armazenados em ambos os sistemas.

Segundo Oliveira e Marcelino (2012), a transformação a ser realizada pelo *gateway* pode ser decidida pelo administrador do projeto utilizando as seguintes estratégias:

- *Forward* ou *Database First*: Os dados são direcionados do sistema legado para uma nova estrutura de modelo de dados. O *forward*, denota que o sistema de origem envia os dados para o destino.
- *Reverse* ou *Database Last*: O sistema de destino busca a informação intermediada pelo *gateway* no sistema de origem.

A Figura 6 mostra a arquitetura do processo de migração conforme especificado.

Figura 6 – Exemplo de arquitetura da migração *Chicken Little*



Fonte: Brodie (1993).

Os onze passos que envolvem a metodologia segundo Broy (2005) são:

1. Analisar o sistema legado;
2. Decompor a estrutura do sistema legado;
3. Projetar a interface de destino;
4. Projetar a aplicação de destino;
5. Projetar a base de dados destino;
6. Instalar o ambiente destino;
7. Criar e instalar os *gateways* necessários;
8. Migrar as bases legadas;
9. Migrar as aplicações legadas;
10. Migrar as interfaces legadas;
11. Mudar pra o sistema de destino.

Percebe-se pelos passos envolvidos que a *Chicken Little* ainda significa uma reformulação completa do sistema (BROY, 2005). Essa estratégia incremental inicia seu processo com partes do sistema que são independentes, e gradualmente valida se cada passo da migração teve êxito. Executando dessa maneira garante-se uma segurança durante o processo de migração, pois se um passo em uma fase avançada do processo falhar, é possível retomar a migração do passo anterior (BRODIE, 1993).

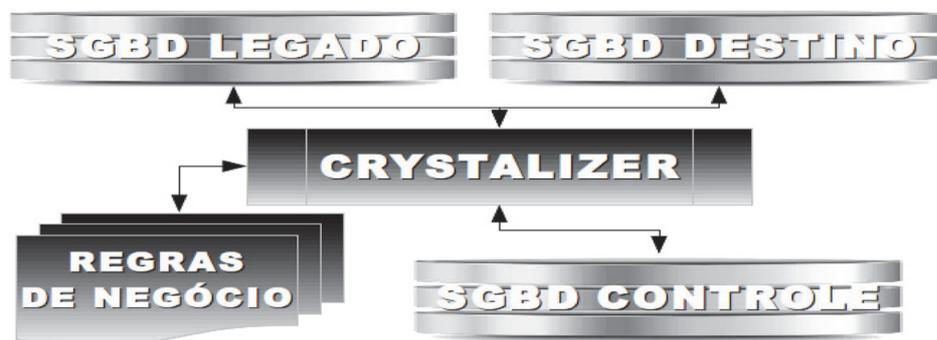
2.3.3 Butterfly

Diferentemente da metodologia *Chicken Little*, esta é uma metodologia que não utiliza a interoperabilidade entre as aplicações, isto é, o sistema legado permanece em operação, enquanto que, em um ambiente de teste, o novo sistema pode ser desenvolvido e testado previamente a uma migração, sem afetar o funcionamento ou mesmo causar transtornos (OLIVEIRA E MARCELINO, 2012).

Sua abordagem é muito semelhante à ETL, com a diferença que esta abordagem propõe o desenvolvimento de um sistema intermediário de migração de

dados chamado *Crystalizer*, que tem como função converter o modelo do banco de dados, transpondo a informação e realizando os ajustes necessários, de um SGBD para o SGBD destino, por meio de regras de negócio – campos que não podem ser nulos ou que serão completados por uma consulta no destino – incluídas manualmente pelo administrador do banco de dados (OLIVEIRA E MARCELINO, 2012). O processo citado pode ser visualizado na Figura 7.

Figura 7 – Exemplo de arquitetura da migração *Butterfly*



Fonte: Oliveira e Marcelino (2012).

Segundo Bing Wu et al. (1997), os passos que compõem a metodologia são:

1. Planejar a migração;
2. Entender a semântica do sistema legado e desenvolver o esquema de dados destino;
3. Migrar todos os componentes, exceto os dados, do sistema legado para a arquitetura destino;
4. Gradualmente migrar os dados legados para o sistema destino e treinar os usuários a usar a aplicação destino; e
5. Passar a utilizar a aplicação destino.

2.3.4 Comparativo

A Tabela 2 compara as metodologias segundo os seguintes critérios: i)

Necessidade de criação de *gateways* ou desenvolvimento de motor de migração durante o processo; ii) se o sistema legado precisa estar fora do ar, sem acesso, para realizar a migração; iii) o porte, tamanho, ao qual a estratégia é indicada, podendo ser pequeno, médio ou grande; iv) se existe a interoperabilidade entre as aplicações durante o processo de migração de dados.

Tabela 2 – Comparativo entre as metodologias ETL, *Chicken Little* e *Butterfly*

Características	ETL	<i>Chicken Little</i> (Forward)	<i>Chicken Little</i> (Reverse)	<i>Butterfly</i>
Necessidade de desenvolvimento/ <i>gateways</i>	Conforme a necessidade	Sim	Sim	Sim
Sistema legado fora do ar	Sim	Não	Não	Sim
Migração de dados primeiro	Sim	Não	Sim	Sim
Porte da migração	Pequeno/Médio	Médio/Grande	Médio/Grande	Pequeno/Médio
Interoperabilidade entre as aplicações	Não	Sim	Sim	Não

Fonte: Autor (2016).

Analisando as informações da tabela, podem-se observar as seguintes informações: i) apenas a metodologia ETL requer, quando necessário, o desenvolvimento de um sistema intermediário, o que contribui para sua baixa complexidade; ii) em caso de migração de grande porte, ou necessidade de não se retirar o sistema do ar ou interoperabilidade entre o sistema legado e o destino deve-se utilizar a metodologia *Chicken Little*.

2.3.5 Trabalhos Relacionados

Foram encontrados um conjunto de trabalhos durante a realização deste estudo. Um projeto de migração incremental, utilizando a metodologia *Chicken Little* de um ERP (*Enterprise Resource Plannig*) denominado SGI (Sistema de Gestão Integrada) foi abordado por Igawa (2013). Os resultados obtidos, segundo Igawa

(2013), quanto a abordagem incremental proposta se mostrou uma estratégia muito interessante visto o sistema ser de grande porte. O fato dos passos de migração serem efetuados em áreas específicas do sistema ofereceu maior segurança durante o processo, visto o sistema possuir centenas de classes (IGAWA, 2013).

Mendonça (2009) considerou as metodologias *Chicken Little* e *Butterfly*, porém por encontrar lacunas em suas técnicas, criou uma nova metodologia de migração de dados com seis etapas e cinquenta e duas atividades. Entretanto, o projeto era de porte grande e foi extenso, durando cerca de dezoito meses, com uma equipe de vinte pessoas atuando diretamente com a metodologia em suas atividades (MENDONÇA, 2009). Por ser uma metodologia nova, certas melhorias foram encontradas e não aplicadas, além disso a metodologia não foi testada em um caso de migração de um sistema de porte pequeno (MENDONÇA, 2009).

Oliveira e Marcelino (2012) abordaram também as três metodologias apresentadas nesta seção e dentre elas observaram que as metodologias ETL e *Butterfly* possuem os mesmos princípios e recomendações, enquanto a metodologia *Chicken Little* é mais burocrática, pois o processo de migração é dividido em diversas partes. A migração incremental pode ser interessante quando existe uma equipe de trabalho com a possibilidade de distribuir melhor as funcionalidades de controle de cada camada (OLIVEIRA E MARCELINO, 2012).

Além disso, Oliveira e Marcelino (2012) concluíram que embora as metodologias sejam genéricas, podem ser usadas como referência ao realizar atividades como integração de sistemas e atualização de estrutura e migração de dados em sistemas legados. Os autores ainda salientam que cada sistema tem seus pontos fortes e fracos em seu desenvolvimento e cada caso é um caso necessitando de uma análise.

2.4 FERRAMENTAS DE MIGRAÇÃO DE DADOS

Existem ferramentas que possuem funcionalidades que auxiliam em uma

migração de dados mais prática e ágil. Segue ferramentas analisadas:

1. **DBTool Manager Professional**⁴: é uma ferramenta *freeware* para gerenciamento de banco de dados com muitas funcionalidades e com suporte aos principais SGBD conhecidos, como Oracle, MySQL, SQL Server, PostgreSQL, entre outros. A versão *freeware* possui apenas uma restrição de importar até 1.000.000 de volume das tabelas. Esta ferramenta dispõe de bons recursos de administração de banco de dados, que possibilitam a geração de *script* DDL com os comandos SQL de criação de todo o banco, inclusive seus objetos e com a vantagem deste roteiro já estar totalmente compatível para o PostgreSQL.
2. **JExodus** (Rodrigues-Neto e Passos, 2009): Ferramenta desenvolvida para realizar a migração de bases de dados tendo como fonte de dados um SGBD, documentos de texto ou objetos Java, de forma que o desenvolvedor efetue a migração de dados sem se preocupar com detalhes de baixo nível como conexão com banco de dados, comandos SQL ou implementar código para ler dados de diversas fontes. Possui reengenharia de dados a partir da criação de classes Java que especifiquem a rotina de transformação necessária. Também permite que o esquema de dados destino seja alterado, porém isso se dá via arquivos de configuração que detalham as modelagens – antiga e nova – a serem consideradas. Devido o seu formato intensamente baseado em arquivos XML o uso dessa ferramenta pode ser mais oneroso que as demais.
3. **DBConvert**⁵: Ferramenta da empresa DMSOft Technologies, possui diversas opções de migração entre SGBDs diferentes. Entretanto, a empresa não possui a opção de MySQL para PostgreSQL.
4. **Swissql**⁶: Ferramenta comercial da empresa Zoho, é capaz de realizar a migração de dados entre SGBDs diferentes e também através de arquivos

4 <http://www.dbtools.com.br/EN/dbmanagerpro/>

5 <http://dbconvert.com>

6 <http://info.swissql.com>

(cvs, txt). A ferramenta oferece suporte para os SGBDs Oracle, IBM DB2, SQL Server, Sybase, MySQL MaxDB, MySQL, PostgreSQL e MS Access databases. A ferramenta em sua versão trial, permite 30 dias de testes e possui algumas limitações no número de tabelas, linhas e índices.

2.5 CONSIDERAÇÕES FINAIS

Nos estudos de caso observamos que a metodologia *Chicken Little* é burocrática e divide o processo de migração em diversas partes, o que o torna como um todo mais lento, porém mais seguro para sistemas de grande porte. Como resultado desta análise, as diretrizes apontadas pela metodologia ETL, por se tratar de uma abordagem mais adequada a migração de pequeno porte, servem de base para o desenvolvimento da migração de dados do portal administrativo do IntergenicDB.

Visto que o SGBD MySQL possui semelhanças com o SGBD PostgreSQL, pode-se realizar a migração de dados gerando os *scripts* SQL – criação das tabelas, índices, entre outros e inserção dos dados – da base de dados MySQL. Após análise dos *scripts*, alterando-os em caso de necessidade, pode-se executá-los no PostgreSQL através da ferramenta PGAdmin.

3 PROPOSTA DE SOLUÇÃO

O processo de evolução de software inicia com uma solicitação de mudança (SOMMERVILLE, 2011). As modificações que serão realizadas no portal fazem parte deste processo evolutivo. Além disso, conforme apresentado na seção anterior, a evolução de software pode envolver uma etapa chamada reengenharia de *software*. Será aplicada uma das atividades da reengenharia de *software*, a reestruturação de dados, durante a migração de dados do portal administrativo do Intergenidb.

O presente capítulo apresentará a análise da migração de dados a ser efetuada no portal. Serão apresentadas as demais modificações, e se necessário, as ferramentas que serão utilizadas para o desenvolvimento dessas modificações.

3.1 MIGRAÇÃO DE DADOS

O portal administrativo do Intergenidb 1.0 (vide o Apêndice A) utiliza o SGBD MySQL. Entretanto, este possui algumas limitações que estão dificultando os usuários do portal realizar atividades como: dificuldade na criação de *scripts* SQL compatíveis com o MySQL, para consultar os dados diretamente no servidor do banco de dados, visto que nem todos os padrões das versões mais atuais do SQL foram importados para o MySQL, lentidão nas consultas dos dados por meio da interface e erros na exportação dos dados do banco por meio da ferramenta phpMyAdmin. Em virtude de tais problemas, optou-se pela migração dos dados para o SGBD PostgreSQL.

A migração dos dados do portal administrativo deverá manter a integridade dos dados, evitar dados repetidos ou perdas. A proposta migração dos dados que contemplam a parte administrativa do portal seguirá a metodologia de ETL, apresentada na seção anterior. Com o estudo das metodologias e dos trabalhos relacionados, pode-se entender que a ETL é apropriada para migrar os dados do portal administrativo do Intergenidb devido ao fato de a migração de dados não ter

a necessidade de ser incremental, de o sistema poder ficar fora do ar por um determinado tempo e o método de extração, transformação e carga ser um processo de baixa complexidade.

A parte administrativa do portal possui 12 tabelas (ANEXO A) (INTERGENICDB, 2015). O sistema não utiliza *triggers*, procedimentos ou *views* nesta área de gerenciamento das informações. Os demais objetos do banco, que atendem a área de acesso ao público, será migrado em outro momento. A seguir será detalhado cada uma das ações que serão realizadas na migração: extração, transformação e carga.

3.1.1 Extração

A extração de dados se dará por meio da ferramenta phpMyAdmin junto ao *site* de hospedagem do portal do IntergenicDB. A ferramenta permite escolher as tabelas e os dados que serão exportados. O phpMyAdmin gera um arquivo *.sql* com os comandos de criação de tabela, índices e inserção de dados. O arquivo será analisado e será preciso fazer algumas alterações. Na tela de exportação que o phpMyAdmin disponibiliza escolher o método de exportação “rápida” ou “customizado”. A opção rápida permite apenas escolher o formato do arquivo que será gerado, enquanto que a customizado permite configurar algumas opções para geração do arquivo, como por exemplo: se deve ser incluída a linha de comando de teste se a tabela já existe no banco atual, se deve se incluir o comando de apagar a tabela/view/procedure, dentre outras opções. Pode-se gerar o arquivo nos formatos *.sql*, *.xls*, *.csv*, *.txt*, *.zip*, entre outros.

3.1.2 Transformação

Uma alteração que ocorrerá na migração da base de dados em todas as tabelas existentes será a alteração da coluna *id* de cada tabela, atualmente do tipo

INT com a propriedade AUTO_INCREMENT para o padrão do PostgreSQL tipo BIGSERIAL. Ao indicar esse tipo para uma coluna, automaticamente é criada uma sequência, utilizando o seguinte padrão: *nometabela_nomecoluna_seq*. A Figura 8 exemplifica o *script* gerado para o MySQL (A) e para o PostgreSQL (B), respectivamente.

Figura 8 – Comparação entre *script* MySQL e PostgreSQL

A	B
<pre>CREATE TABLE `Country` (`CountryId` INT(11) NOT NULL AUTO_INCREMENT, `Name` VARCHAR(512) NOT NULL, PRIMARY KEY (`CountryId`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;</pre>	<pre>CREATE TABLE Country (CountryId BIGSERIAL NOT NULL, Name VARCHAR(512) NOT NULL, PRIMARY KEY (CountryId));</pre>

Fonte: Autor (2016).

A estrutura atual possui uma tabela com o nome “*user*”, entretanto essa é uma palavra privada no PostgreSQL. Por este motivo a tabela será criada na nova base de dados com o nome “*users*”. Será realizada também alterações no código fonte do sistema para que o sistema continue realizando as operações de leitura (*select*), inserção (*insert*), alteração (*update*) e remoção (*delete*) com êxito.

3.1.3 Carga

A carga será realizada por meio da ferramenta PGAdmin3, no *site* de hospedagem, executando o *script* gerado no phpMyAdmin e alterado conforme o item 3.1.2. As operações realizadas envolve a criação das tabelas, ajustes de restrições de integridade e carga de dados por meio do comando *insert*. Após a carga será realizado testes para validar a migração de dados.

3.2 REALIZAÇÃO DE TESTE

Para validar a migração de dados, serão executados os seguintes testes,

acessando o portal Administrativo do IntergenicDB:

- a) Efetuar o *login*;
- b) Acessar as telas de listagem dos cadastros de Usuários, Grupos, Grupos Usuários, Cidades, Países, Publicações, Estados e Conexão;
- c) Executar o filtro na tela de listagem dos cadastrados listados acima que possuam filtro;
- d) Efetuar o cadastro de um novo registro em cada um dos cadastros;
- e) Efetuar a exclusão de um registro.

Como resultado dos testes, deve-se obter a mesma quantidade de registros em cada tela de listagem, bem como os dados devem estar mostrados com a mesma formatação. O objetivo de filtrar, cadastrar um novo registro e efetuar a exclusão será mostrar que as funcionalidades da ferramenta permaneceram intactas, e que apenas a camada de conexão dos dados da aplicação foi alterada.

3.3 CRIAÇÃO DE WEB SERVICE PARA FORNECER DADOS DE *LOGIN*

Web service é um padrão do *World Wide Web Consortium* – W3C utilizado para que aplicativos se comuniquem via internet utilizando tecnologias programáveis e reutilizáveis que aproveitam a flexibilidade da Internet, mesmo que em plataformas diferentes, integrando as informações entre si.

3.3.1 SOAP

Segundo Sommerville (2011), existem três padrões, baseados em XML, fundamentais que possibilitam comunicações entre *Web services* são eles:

1. SOAP (*Simple Object Access Protocol*): Que define um padrão para a troca de dados entre *Web services*;
2. WSDL (*Web Services Description Language*): Que atua no modo de como as interfaces dos *Web services* serão representadas;

3. UDDI (*Universal Description, Discovery and Integration*): É o padrão que está relacionado à descoberta de serviços.

Os *Web services* são projetados e implementados (Figura 9) na linguagem WSDL e publicados em um registro de acesso geral usando o padrão UDDI, assim o solicitante de serviço, que deseja utilizar um serviço, busca o registro UDDI desse serviço para descobrir a especificação e localizar o provedor de serviços. Para se comunicar ao solicitante geralmente utiliza-se o protocolo SOAP (SOMMERVILLE, 2011).

Figura 9 – Arquitetura orientada a serviços



Fonte: Sommerville (2011).

Uma mensagem SOAP é regida pelo esquema XML: <http://www.w3.org/2001/12/soap-envelope>. Toda mensagem SOAP é formada por um elemento envelope, que deve conter os subelementos: *Header* e *Body*. Dentro do *Header* temos informações de controle e dentro do *Body* o corpo da mensagem em si.

3.3.2 REST

REST (Transferência de Estado Representacional) é um estilo de arquitetura para sistemas multimídias distribuídos que enfatiza a generalização das interfaces, a

escalabilidade da integração entre os componentes e a instalação independente dos mesmos (FIELDING, 2000). Segundo Fielding (2000), um estilo arquitetural é um conjunto de regras que restringe os papéis/funcionalidades dos elementos arquiteturais e o grau de relacionamento entre esses elementos nas arquiteturas que aplicam esse estilo. Definir estilos de arquitetura de *software* é um recurso que permite nomear um conjunto de padrões, técnicas e regras de forma que seja fácil identificá-los, analisar melhoramentos e aplicá-los de forma efetiva no desenvolvimento de projetos de *software* (GONÇALVES E SILVA, 2012).

Ao contrário de SOAP, que é por si só um protocolo de envio e recebimento de mensagens, REST não define como as mensagens são enviadas e recebidas. Ao invés disso, ele utiliza o próprio protocolo HTTP (GONÇALVES E SILVA, 2012). O protocolo HTTP possui métodos uniformes (GET, POST, PUT e DELETE), onde cada recurso pode ter uma ou mais representação (XML, JSON, Text, ect) as quais são transferidas entre o cliente e o serviço, durante a invocação ao mesmo. A Tabela 3 relaciona os métodos HTTP, com a ação, sua equivalência em SQL (*Structured Query Language*) e descreve a funcionalidade de cada item (RICHARDSON e RUBY, 2007).

Tabela 3 – Relacionamento entre verbos SQL e HTTP

Método HTTP	Ação	SQL	HTTP
POST	Create	INSERT	Cria um novo recurso, inserido dados
GET	Read	SELECT	Obtém os dados de um recurso
PUT	Update	UPDATE	Atualiza os dados de um recurso
DELETE	Delete	DELETE	Exclui um recurso e seus dados

Fonte: Richardson e Rubi (2007).

3.3.3 Resolução

Na plataforma .NET as funcionalidades dos *Web services* podem ser criadas e definidas em arquivos do tipo SVC, utilizando as linguagens C# ou VisualBasic.

Para que as funcionalidades de um *Web service* possam ser acessadas por outras aplicações, é necessário que o arquivo SVC esteja colocado em uma pasta virtual de um servidor *Web* que possua o IIS e a plataforma .NET instalada. Na plataforma .NET, o consumo de *Web services* pode ser realizado por aplicações escritas em qualquer linguagem que suporta o consumo de *Web services*.

O IntergenicDB disponibilizará por meio do estilo de arquitetura REST com retorno em JSON, três operações para que outros sistemas tenham acesso aos dados de usuários sem a necessidade de criar uma conexão com o PostgreSQL. As operações que serão disponibilizadas são: inclusão de usuário, alteração de dados do usuário incluindo a alteração da senha e validação de *login*. A Tabela 4 mostra o formato do endereço *web* de acesso do serviço e o objetivo de cada chamada.

Tabela 4 – Operações disponibilizadas pelo *web service*

ROTA	VERBO HTTP	DESCRIÇÃO
<i>/service/users/</i>	POST	Adiciona um novo usuário
<i>/service/users/:email</i>	PUT	Atualiza os dados de um usuário
<i>/service/login/:usuario:senha</i>	GET	Retorna se o <i>login</i> é válido

Fonte: Autor (2016).

O método utilizando o protocolo POST, que servirá para adicionar um novo usuário, deverá enviar os seguintes parâmetros conforme Tabela 5:

Tabela 5 – Parâmetros do método utilizando protocolo POST

Parâmetros	Tipo	Obrigatório	Observações
<i>firstname</i>	String	Sim	Nome do usuário
<i>lastname</i>	String	Sim	Sobrenome do usuário
<i>email</i>	String	Sim	<i>E-mail</i> do usuário
<i>password</i>	String	Sim	Senha do usuário criptografada com a função MD5
<i>country</i>	String	Sim	País do usuário

Fonte: Autor (2016).

O método utilizando o protocolo PUT poderá enviar os parâmetros especificados na Tabela 6, sendo obrigatório o preenchimento de ao menos para atualização do usuário.

Tabela 6 – Parâmetros do método utilizando protocolo PUT

Parâmetros	Tipo	Obrigatório	Observações
<i>firstname</i>	String	Não	Nome do usuário
<i>lastname</i>	String	Não	Sobrenome do usuário
<i>password</i>	String	Não	Senha do usuário criptografada com a função MD5
<i>country</i>	String	Não	País do usuário

Fonte: Autor (2016).

O método **login** utilizando o protocolo GET retornará se o *login* pode ser efetuado com sucesso ou não. Os parâmetros que precisam ser enviados podem ser observados na tabela Tabela 7.

Tabela 7 – Parâmetros do método utilizando protocolo GET

Parâmetros	Tipo	Obrigatório	Observações
<i>email</i>	String	Sim	<i>E-mail</i> do usuário
<i>password</i>	String	Sim	Senha do usuário criptografada com a função MD5

Fonte: Autor (2016).

3.4 RECUPERAR SENHA

O portal IntergenicDB tem as senhas de seus usuários gravadas no banco de dados criptografadas. Ao tentar recuperar a senha, o sistema envia um *e-mail* para o usuário informando a senha atual. O problema é que ele está enviando a senha criptografada. A criptografia utilizada no sistema não possui descryptografia, por isso a solução será implementar a criação de uma nova senha com letras e números randômicos, no tamanho mínimo de 6 caracteres para o usuário solicitante de

recuperação de senha e enviá-la para o *e-mail* cadastrado e por fim, salvá-la criptografada no banco de dados.

3.5 CADASTRO DE AJUDA

Para facilitar a manutenção da tela de ajuda do portal IntergenicDB (Figura 10) será criado um cadastro na área administrativa que será acessado pelos administradores mediante *login* com usuário e senha. Esse cadastro proporcionará a inclusão, alteração e exclusão de categorias que são os agrupadores dos conteúdos como: *IntergenicDB Models*, *Search*, *Update Files* e *Administrator Activities* e seus respectivos conteúdos, conforme existe atualmente (Figura 10).

Figura 10 – Tela de ajuda do portal do IntergenicDB



IntergenicDB
Prokaryotes' Intergenic Regions

Início | Pesquisar | Publicações | **Ajuda**

IntergenicDB Models

- [Flowchart \(scientific workflow\)](#)
- [PromotoresDB database model](#)
- [IntergenicDB database model](#)

Search

- [Organism](#)
- [Gene](#)
- [Intergenic Region](#)

Update Files

- [File Format to be uploaded](#)

Administrator Activities

- [Group and user management](#)
- [Molecular biology management](#)
- [Country management](#)
- [Publication management](#)
- [Import data from uploaded files](#)

Entrar

Usuário:

Senha:

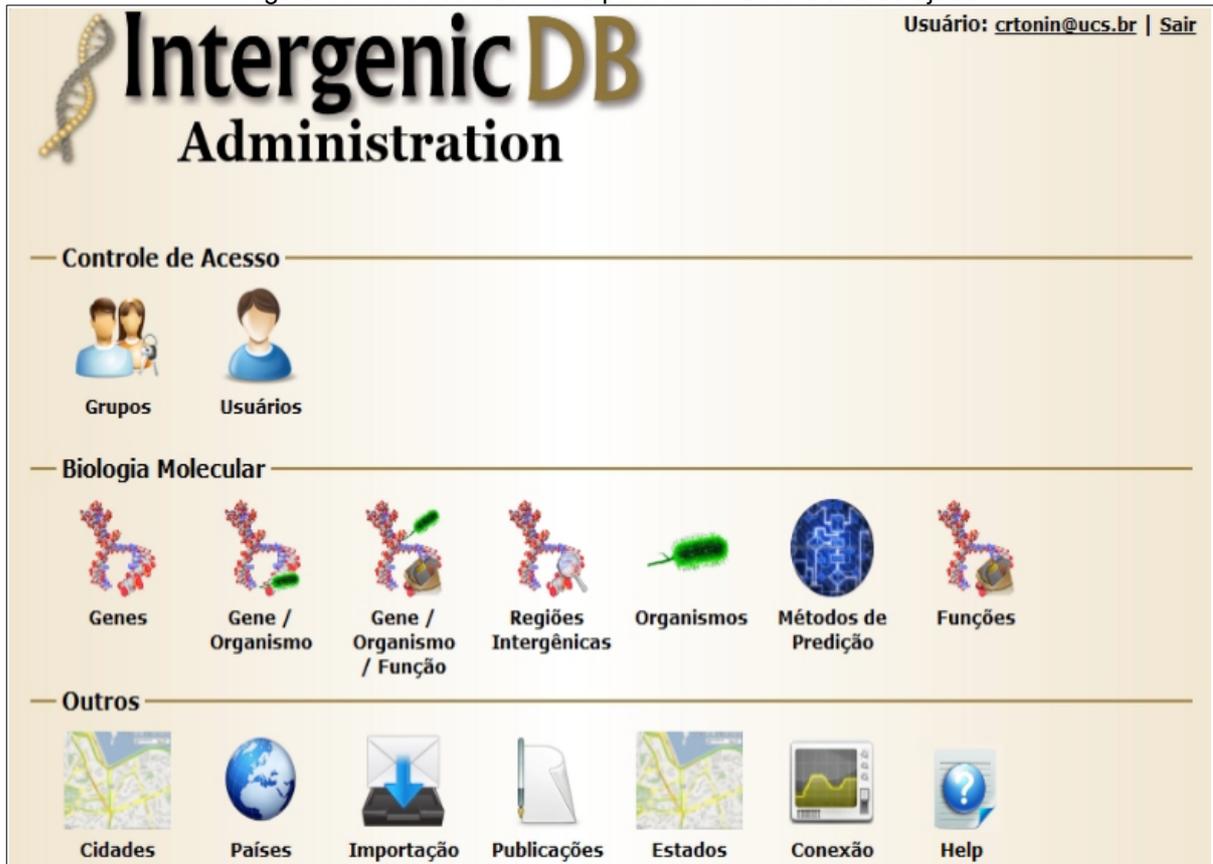
[Ainda não está registrado?](#)
[Esqueceu sua senha?](#)

Fonte: IntergenicDB⁷ (2016).

7 Disponível em: <<http://intergenicdb.bioinfocps.com/Default/Help>>. Acesso em: 08 jun 2016.

O menu de acesso “Help” referente ao novo cadastro será incluído em Outros, ao lado da funcionalidade Conexão. O novo menu pode ser visualizado na (Figura 11).

Figura 11 – Inclusão do menu para administrar a tela de ajuda



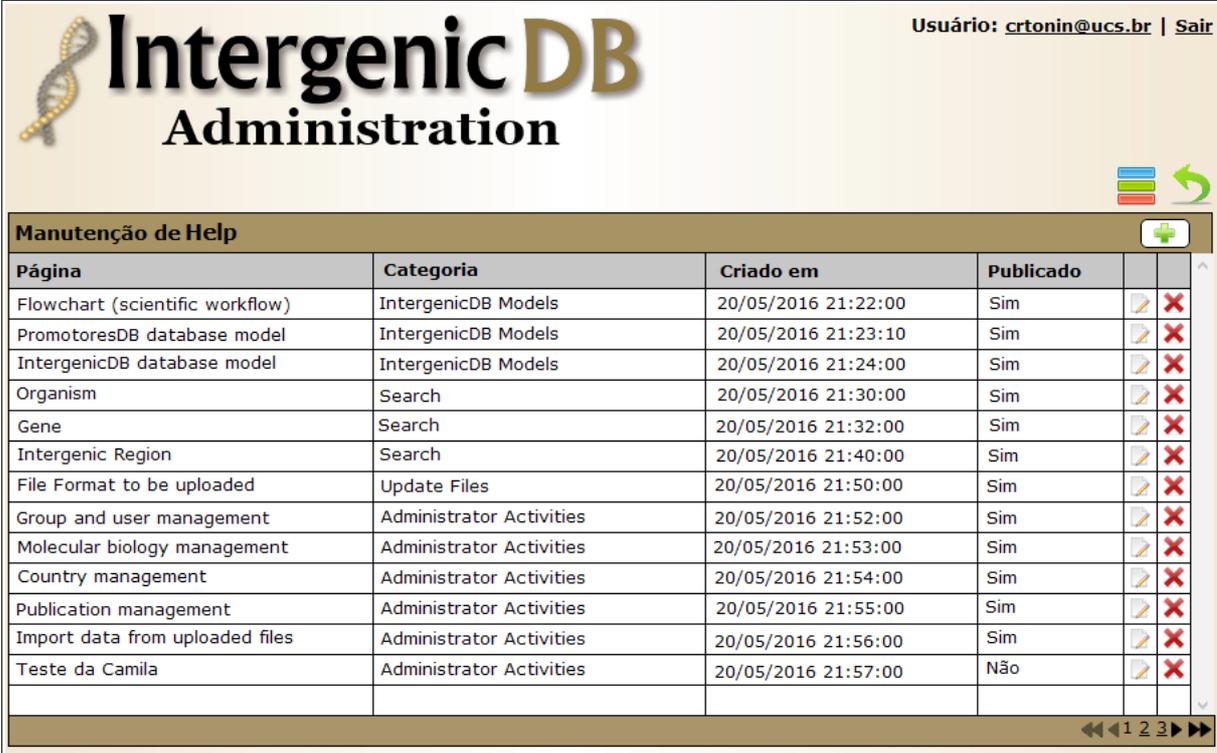
Fonte: Autor (2016).

O cadastro de ajuda conterà um título, a vinculação a uma categoria criada anteriormente, a opção de um editor *HTML* que permitirá a inclusão de texto e imagens. Será possível informar os conteúdos em português e inglês. Haverá um campo publicado, com as opções Sim e Não permitindo que usuário defina quando a página poderá ser acessada no *site*.

Seguindo o padrão já existente, a tela de listagem das categorias terá as funções de inclusão, edição e exclusão. O cadastro (Figura 13) de categorias permitirá o usuário informar o nome da categoria em 2 idiomas: português e inglês.

Atualmente o portal só exibe as categorias em inglês.

Figura 12 – Tela de listagem do novo cadastro de ajuda

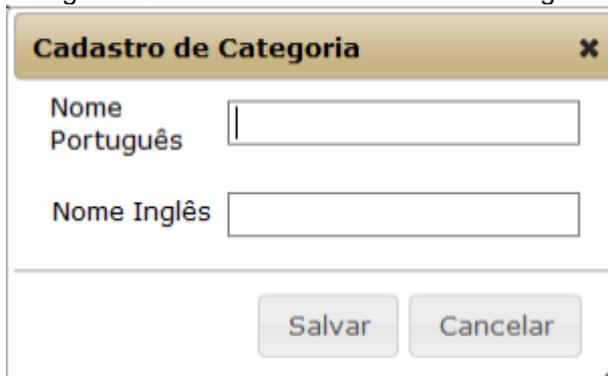


The screenshot shows the 'IntergenicDB Administration' interface. At the top right, the user is identified as 'Usuário: crtonin@ucs.br | Sair'. Below the header is a table titled 'Manutenção de Help' with columns for 'Página', 'Categoria', 'Criado em', and 'Publicado'. The table lists 13 entries, each with a 'Publicado' status of 'Sim' or 'Não' and a 'Teste da Camila' status of 'Sim' or 'Não'.

Página	Categoria	Criado em	Publicado		
Flowchart (scientific workflow)	IntergenicDB Models	20/05/2016 21:22:00	Sim		
PromotoresDB database model	IntergenicDB Models	20/05/2016 21:23:10	Sim		
IntergenicDB database model	IntergenicDB Models	20/05/2016 21:24:00	Sim		
Organism	Search	20/05/2016 21:30:00	Sim		
Gene	Search	20/05/2016 21:32:00	Sim		
Intergenic Region	Search	20/05/2016 21:40:00	Sim		
File Format to be uploaded	Update Files	20/05/2016 21:50:00	Sim		
Group and user management	Administrator Activities	20/05/2016 21:52:00	Sim		
Molecular biology management	Administrator Activities	20/05/2016 21:53:00	Sim		
Country management	Administrator Activities	20/05/2016 21:54:00	Sim		
Publication management	Administrator Activities	20/05/2016 21:55:00	Sim		
Import data from uploaded files	Administrator Activities	20/05/2016 21:56:00	Sim		
Teste da Camila	Administrator Activities	20/05/2016 21:57:00	Não		

Fonte: Autor (2016).

Figura 13 – Tela do novo cadastro de categoria



The screenshot shows a 'Cadastro de Categoria' dialog box. It contains two text input fields: 'Nome Português' and 'Nome Inglês'. Below the fields are two buttons: 'Salvar' and 'Cancelar'.

Fonte: Autor (2016).

3.6 TROCA DE SISTEMA DE CONTROLE DE VERSÕES

O Sistema de Controle de Versões (SCV), de forma simples, é um local para armazenamento de artefatos gerados durante o desenvolvimento de sistemas de software (MASON, 2006). Por meio do SCV é possível obter históricos das modificações, permitir que vários desenvolvedores trabalhem no mesmo projeto e permite um comparativo entre várias versões do projeto.

Existem muitos sistemas de controle de versões disponíveis, cada um com características e funcionalidades particulares. Atualmente o IntergenicDB utiliza o Subversion (SVN). O forte crescimento da Internet e dos ambientes de colaboração, apesar dos avanços significativos obtidos pelo SVN, exigiram ferramentas que atendessem as demandas de grandes comunidades de desenvolvedores espalhadas pelo mundo.

O GIT foi inicialmente desenvolvido com o intuito de criar uma ferramenta para controlar o desenvolvimento do *kernel* do Linux, com foco em usuários do Linux e bom desempenho neste sistema operacional. Apesar disso, o GIT também pode ser usado em outros sistemas operacionais baseados na plataforma Unix e no Windows.

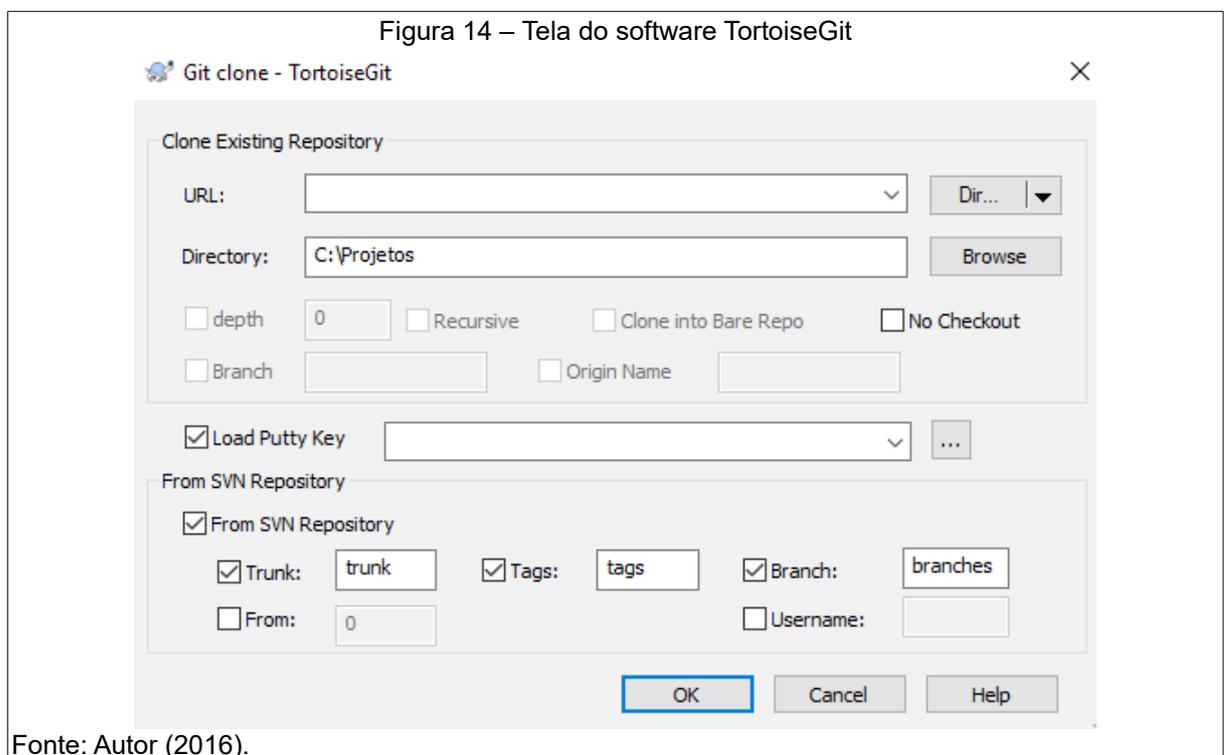
Para gerenciar as informações geradas pelo SCV existem ferramentas que possuem painéis de controle. Os *sites* de hospedagem de projetos além dos painéis de controle, possuem *wiki*, fóruns para problemas, entre outros benefícios. Caso seu repositório seja público, é permitido que pessoas anônimas acessem o conteúdo de código aberto, podendo copiar ou criar uma derivação e contribuir com o autor original. Além disso, é possível utilizar os repositórios em modo privado entre colaboradores previamente cadastrados.

O GitHub⁸, um dos mais conhecidos *sites* de hospedagem de projetos da atualidade, utiliza uma licença paga para repositórios privados, permitindo um número infinito de colaboradores. Seu concorrente, o Bitbucket⁹, entretanto, possui

⁸ <http://github.com/>

⁹ <http://bitbucket.org/>

um plano gratuito para uma equipe de até 5 usuários porém, com infinitos repositórios privados. Por este motivo, será utilizado o Bitbucket como serviço de hospedagem do projeto do IntergeneticDB. O GIT possui comandos que permitem a migração do repositório SVN para o GIT sem perder nenhum histórico até o momento. Será utilizado o software TortoiseGit¹⁰ (Figura 14) para realizar essa migração.



¹⁰ <http://tortoisegit.org/>

4 IMPLEMENTAÇÃO

O atual SGBD do portal administrativo do IntergenicDB é o MySQL. Conforme já considerado, este possui algumas limitações que estão dificultando os usuários do portal realizar algumas atividades principalmente pela lentidão nas consultas dos dados. Devido a tais problemas, optou-se pela migração dos dados para o SGBD PostgreSQL.

O presente capítulo descreve a migração dos dados do SGBD MySQL para PostgreSQL utilizando a metodologia ETL proposta para realização deste trabalho. Será detalhado o processo de implementação das alterações que foram propostas para o portal. Este capítulo segue a mesma organização de seções utilizada no Capítulo 3 que descreveu a proposta de solução.

4.1 MIGRAÇÃO DE DADOS

4.1.1 Migração das tabelas

Para realizar a migração dos dados foi necessário acessar a base MySQL no servidor do *KingHost* onde o portal está hospedado. Por meio da ferramenta phpMyAdmin fornecido pela hospedagem foi gerado os *scripts* contendo a estrutura das tabelas e os dados a serem inseridos no formato SQL. A migração dos dados foi realizada comparando-se as definições das tabelas em MySQL e o PostgreSQL. A tabela *user* foi renomeada para *users* devido ao fato da palavra *user* ser privada no PostgreSQL.

Foi realizado uma comparação entre as colunas de cada tabela utilizando com base a tabela de equivalência de Oliveira e Marcelino (2012) apresentada na seção 2.2 deste trabalho. Para cada tabela foi preciso também atualizar a *sequence* no último valor informado AUTO_INCREMENT. Para isso, foi gerado o script: `SELECT setval('f<nome_sequencia>', <valor>);`

As tabelas *city* e *state* não foram migradas devido ao fato de não serem populadas com dados no sistema. Foi definido que deve ser guardada apenas a informação do país do usuário que acessa o portal, como já está sendo feito atualmente. Na tabela *admin* foi incluída uma coluna *adminid* do tipo BIGSERIAL ¹¹, visto essa tabela não possui uma chave primária definida.. Além disso, a informação da coluna *userid* da tabela *admin* deve ser única, por este motivo, foi incluído a propriedade *unique*.

Na tabela *users* foi incluída uma coluna *upload* do tipo *boolean* que valida se o usuário pode ou não enviar arquivos pela área do *site* após logado. A tabela *publication*, não continha dados, por isso teve sua estrutura alterada para atender melhor as necessidades do usuário. A maioria de suas colunas foram excluídas, permanecendo as colunas *publicationid* e *link*. Foram adicionadas duas novas colunas *description_portuguese* e *description_english* que armazenarão os dados referentes de artigos relacionados.

Foram realizados testes com os acadêmicos Jórdan Rui Rosa e Jovani Dalzochio durante a implementação, percebendo que os dados gerados referente a biologia molecular não estavam corretos. Por isto, as tabelas *geneorganismrole* e *intergenicregion* passaram por adequações em sua estrutura e os dados foram novamente importados pelo Jovani Dalzochio. Foram migradas as seguintes tabelas com a respectiva quantidade de registros conforme a Tabela 8. As demais tabelas do portal foram migradas por ROSA (2016).

Tabela 8 – Tabelas e quantidade de registros que foram importados para o PostgreSQL

Tabela	Quantidade de Registros
Admin	5
AdminGroup	1
Country	5
Users	131

¹¹ Uma coluna do tipo inteiro onde o valor padrão ser atribuído a partir de um gerador de seqüência, semelhante à propriedade AUTO_INCREMENTO existente no MYSQL.

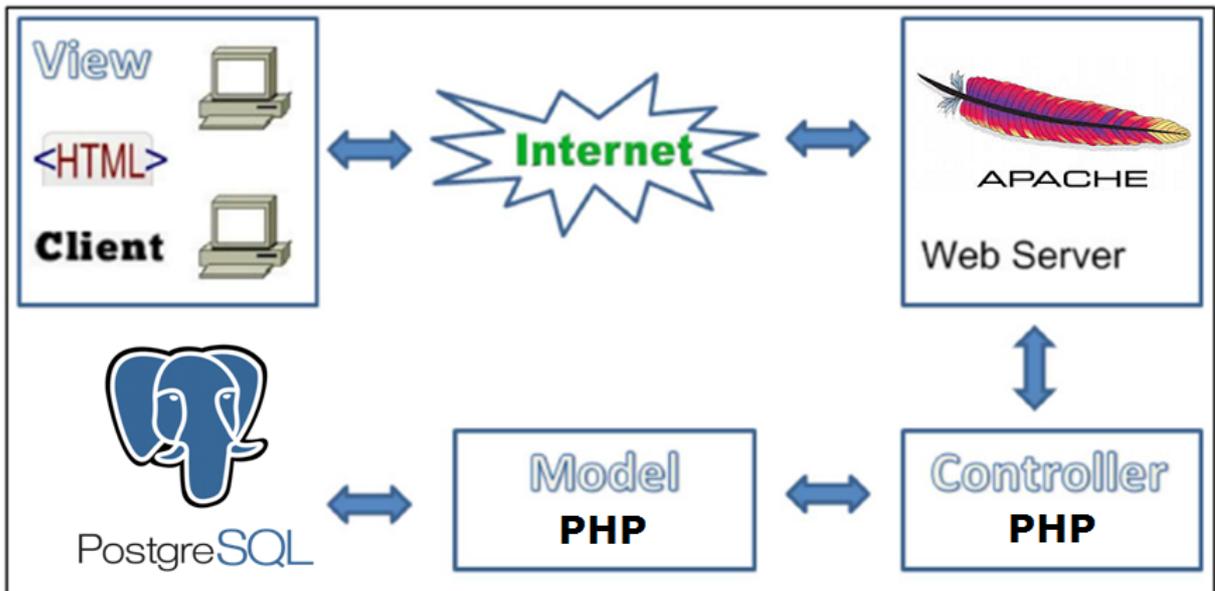
AdminGroupPermission	12
Connection	9079
UserActivationPending	19

Fonte: Autor (2016).

4.1.2 Modificação na arquitetura de software

O portal em sua versão 1.0 foi construído na plataforma .NET com linguagem C#. Foi desenvolvido um *framework* para acessar os dados chamado LP.Framework. Durante o processo de troca de acesso ao banco de dados foi constatado que o LP.Framework e sua maneira de realizar as consultas causa lentidão nos acessos aos dados. A retirada do LP.Framework mostrou-se um trabalho bem extensivo, por este motivo optou-se mudar a plataforma de desenvolvimento do portal, visto que neste mesmo momento a tela de pesquisa também está sendo refeita por ROSA (2016). O portal na versão 2.0 foi construído utilizando a linguagem de programação PHP com padrão de arquitetura MVC, o SGBD PostgreSQL e o servidor web Apache (Figura 15).

Figura 15 – Arquitetura do portal Intergenidb



Fonte: Autor (2016).

Foi utilizado para o desenvolvimento da versão 2.0 do portal o *framework* CodeIgniter¹², *open source* desenhado para o desenvolvimento rápido de projetos. Em conjunto com a troca da linguagem, foi proposto a renovação do *layout* do portal se adequando as tecnologias atuais, possibilitando o uso de um *layout* responsivo – adapta-se aos vários tamanhos de telas disponíveis como celulares, *tablets* ou monitores – mantendo a essência do *layout* atual com a aprovação dos usuários-chave do portal. O *layout* aprovado foi construído utilizando o *framework* CSS *Bootstrap*¹³ e pode ser visualizado na Figura 16.

Figura 16 – Novo *layout* do Portal IntergenicDB



Fonte: Autor (2016).

A área administrativa também teve seu *layout* alterado utilizando os conceitos utilizados no portal, tanto para o *login* como para o acesso aos dados conforme Figura 17. Os itens do menu Biologia Molecular são compostos apenas por tela de consulta, sendo assim, as ações de exclusão e edição não foram implementadas. A

¹² <https://www.codeigniter.com/>

¹³ <http://getbootstrap.com/>

Figura 18 mostra a tela inicial da área administrativa após realizar o *login*.

Figura 17 – Novo *layout* da tela de *login* da área administrativa

Fonte: Autor (2016).

Figura 18 – Tela inicial do portal administrativo do IntergenicDB 2.0

Endereço IP	País	Data de Acesso
177.152.38.181	Brazil	14/11/2016 11:53:52
177.185.196.80	United States	13/09/2016 05:55:17
177.185.196.80	Brazil	13/09/2016 04:31:31
177.185.196.80	United States	13/09/2016 04:03:12
177.185.196.80	United States	13/09/2016 04:03:07

Fonte: Autor (2016).

4.2 CRIAÇÃO DO WEB SERVICE

O *framework* CodeIgniter¹⁴ utiliza os segmentos na URL, de acordo com a abordagem Modelo-Visão-Controle, ou seja: `url-site/classe/funcao/ID` onde a **classe** representa controle que será invocado, **função** representa o método que será chamada. O terceiro, e qualquer outro subsequente, representa o ID ou qualquer outra variável que serão passados para o controle. Então, para criar o *web service* foi necessário criar o controle “*service*” e os métodos *users* e *login*. O retorno do *web service* informará o código de status HTTP que poderá contemplar os informados na Tabela 9.

Tabela 9 – Status HTTP que podem retornar do *web service*

HTTP STATUS CÓDIGO	DESCRIÇÃO
200 (<i>OK</i>)	Executado com sucesso
201 (<i>Created</i>)	Registro criado com sucesso
302 (<i>Found</i>)	Registro encontrado
400 (<i>Bad Request</i>)	Quando não são enviados os parâmetros corretamente para a requisição.
404 (<i>Not Found</i>)	Registro não encontrado
405 (<i>Method not allowed</i>)	Caso alguma requisição seja feita utilizando um método de requisição não tratado, por exemplo DELETE.
500 (<i>Internal Server Error</i>)	Erro ao tentar executar a requisição

Fonte: Autor (2016).

Além disso, retorna uma mensagem codificada em JSON, em caso de sucesso: `{'success:' 'Mensagem de sucesso'}`, enquanto que em caso de erro: `{'error:' 'Erro ocorrido'}`. Para a chamada usuário utilizando o método HTTP POST é validado se todos os campos foram informados: *firstname*, *lastname*, *email*, *password* e *country*. A tabela Tabela 10 descreve os parâmetros e os códigos de

14 <https://www.codeigniter.com/>

retorno possíveis.

Tabela 10 – Parâmetros e retorno do método *users* utilizando POST

ROTA	/service/users	
VERBO HTTP	POST	
PARÂMETROS	Nome	Obrigatório
	<i>firstname</i>	Sim
	<i>lastname</i>	Sim
	<i>email</i>	Sim
	<i>password</i>	Sim
	<i>country</i>	Sim
HTTP STATUS CÓDIGO	201 – Usuário [e-mail] criado com sucesso	
	400 – O campo [campo] é obrigatório.	

Fonte: Autor (2016).

Para chamada usuário utilizando o método HTTP PUT é validado se o *e-mail* e ao menos uma outra informação foram enviadas. A Tabela 11 descreve os parâmetros e os *status* HTTP códigos de retorno possíveis.

Tabela 11 – Parâmetros e retorno do método *users* utilizando PUT

ROTA	/service/users	
VERBO HTTP	PUT	
PARÂMETROS	Nome	Obrigatório
	<i>firstname</i>	Não
	<i>lastname</i>	Não
	<i>email</i>	Sim
	<i>password</i>	Não
	<i>country</i>	Não
HTTP STATUS CÓDIGO	200 – Usuário [e-mail] atualizado com sucesso!	
	400 – Número de parâmetros insuficientes.	

	404 – Usuário [<i>e-mail</i>] não encontrado.
	500 – Ocorreu um erro ao tentar atualizar o usuário [email]

Fonte: Autor (2016).

A requisição “*login*” utilizando o método HTTP GET retornará o *status* HTTP código 302, caso encontre um usuário com as informações de *e-mail* e senha enviados e a mensagem de retorno: “Usuário encontrado”. Caso contrário será retornado o *status* HTTP código 500 e a mensagem de erro será: “Usuário e/ou senha incorretos”. Se o método “*login*” for requisitado sem informar o *e-mail* e/ou senha será retornado o *status* HTTP código 400 e a mensagem de erro será: “Número de insuficientes de parâmetros”. A Tabela 12 descreve os parâmetros e os retornos possíveis.

Tabela 12 – Parâmetros e retorno do método *login* utilizando GET

ROTA	/service/login	
VERBO HTTP	GET	
PARÂMETROS	Nome	Obrigatório
	<i>email</i>	Sim
	<i>password</i>	Sim
HTTP STATUS CÓDIGO	302 – Usuário encontrado	
	500 – Usuário e/ou senha incorretos.	
	400 – Número de insuficientes de parâmetros	

Fonte: Autor (2016).

4.3 RECUPERAR SENHA

Nas páginas do *site*, foi incluída a opção de recuperar a senha conforme Figura 19. Ao clicar nesse link e informar o *e-mail* de acesso ao portal é enviado um *e-mail* com uma nova senha gerada automaticamente com 6 caracteres, contendo

letras, números e caracteres especiais.

Figura 19 – Área do *site* para acessar o link para recuperar a senha

Fonte: Autor (2016).

4.4 CADASTRO DE AJUDA

O cadastro de ajuda foi criado na área administrativa do portal possibilitando o usuário gerenciar os textos de ajuda que serão apresentados na aba Ajuda do *site*. Foi utilizado o editor de texto *html* Summernote¹⁵ que permite o usuário utilizar as opções de texto negrito, itálico, inclusão de *link*, a inclusão de imagens, entre outras. Cada registro de ajuda criado pertence a uma categoria que também é gerenciada pelo usuário por meio do novo cadastro de Categorias.

4.5 TROCA CONTROLE DE VERSÕES

A troca do controle de versões do SVN para o GIT foi realizada executando o comando `git svn` fornecido pelo próprio GIT. Para que os históricos não fossem perdidos foi seguida a documentação do GIT e criado de um arquivo texto com os dados de todos os usuários que já haviam realizados *commit* pelo SVN.

Após configurado o arquivo de usuários, o comando abaixo foi executado: `svn git svn://bioinfoucs.com/bioinfoucs/intergenicdb/ --no-minimize-url --verbose --username bioinfoucs --authors <caminho_arquivo>\users.txt`. Foi criada uma conta

¹⁵ <http://summernote.org/>

no Bitbucket¹⁶ e após por meio da ferramenta GitTortoise foi realizado o procedimento de *push*, ou seja, envio do código fonte ao servidor do novo repositório.

Devido ao fato de o projeto ter sido reformulado de C# para o PHP foi criado um repositório chamado “Intergenicdb 1.0” para armazenar o código-fonte em .NET para fins de consulta. Foi criado um repositório chamado “Intergenicdb” para armazenar o novo código-fonte em PHP.

4.6 NOVAS FUNCIONALIDADES

Devido a versão 2.0 do IntergenicDB ter sido totalmente implementada novamente utilizando outras tecnologias, foram implementadas também novas funcionalidades para ajudar os usuários administradores e os usuários que acessam a área do *site*.

4.6.1 Texto página inicial do *site*

O cadastro do texto da tela inicial permite modificar o texto em português e inglês que será mostrado conforme o idioma do usuário no portal. Para armazenar as informações desse cadastro foi criada uma tabela chamada “*text_home*”. Também foi utilizado o editor de texto *HTML Summernote*¹⁷ que permite o usuário utilizar as opções de texto negrito, itálico, inclusão de *link*, entre outras.

4.6.2 Meus dados e Troca de Senha

A versão anterior do portal não permitia ao usuário da área do *site* modificar seus dados ou trocar a sua senha. A versão 2.0 permite o usuário logado alterar seus dados e trocar sua senha. Essas opções ficam visíveis ao usuário após ser

16 <https://bitbucket.org/>

17 <http://summernote.org/>

efetuado o *login* na área do *site* (Figura 20), abaixo do seu nome no canto direito da página. Na área administrativa do *site* também foram implementadas essas funcionalidades.

Figura 20 – Novas funcionalidades para o usuário logado



Fonte: Autor (2016).

4.6.3 Cadastro de Publicações

O cadastro de publicações teve a alteração de ser um campo único de descrição, fornecendo assim mais liberdade para o usuário, já que as informações referentes a publicações podem variar de um artigo, para um trabalho de conclusão ou um capítulo de livro. Os campos que existiam na versão 1.0 limitavam as informações que o usuário podia informar.

4.6.4 Formulário de Contato

Foi incluída uma nova aba na área do *site* com um formulário para facilitar o contato dos usuários com os administradores do portal IntergenicDB. O formulário possui os campos de nome, *e-mail* e mensagem e todos os campos são obrigatórios para o envio do formulário. Os dados de contato informados são enviados por *e-mail* para os administradores que poderão entrar em contato posteriormente.

4.6.5 Cadastro de Grupo de Administradores

O cadastro de grupo de administradores (Figura 21) teve a inclusão da liberação dos novos cadastros, textos e publicações, e a alteração nas permissões do menu biologia molecular podendo liberar o acesso apenas para visualização, visto as funcionalidades de edição e exclusão terem sido removidas.

Figura 21 – Permissões do cadastro de grupos

Permissões:				
Menu	Visualizar	Criar	Editar	Excluir
Cadastros	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Grupos	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Usuários	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Países	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Biologia Molecular	<input type="checkbox"/>			
Genes	<input checked="" type="checkbox"/>			
Funções	<input checked="" type="checkbox"/>			
Famílias	<input checked="" type="checkbox"/>			
Reinos	<input checked="" type="checkbox"/>			
Organismos	<input checked="" type="checkbox"/>			
Gene/Organismo	<input checked="" type="checkbox"/>			
Gene/Organismo/Função	<input checked="" type="checkbox"/>			
Regiões Intergênicas	<input checked="" type="checkbox"/>			
Métodos de Predição	<input checked="" type="checkbox"/>			
Publicações	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Acessos	<input checked="" type="checkbox"/>			
Textos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Home			<input checked="" type="checkbox"/>	
Ajuda	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Categoria (Texto Ajuda)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fonte: Autor (2016).

4.6.6 Cadastro de Usuários

Na versão 1.0 do portal IntergenDB quando o usuário realizava o *login* a aba que permitia o envio de um arquivo ficava visível. Na versão 2.0 o usuário apenas poderá acessar essa aba se tiver a permissão que os administradores do portal liberarem por meio do cadastro de usuários na área administrativa do portal.

Figura 22 – Tela de cadastro de usuário na área administrativa

Usuário

Nome:

Sobrenome:

Tipo:

Grupo:

Telefone:

País:

E-mail:

Nova Senha:

Ativo
 Upload

Fonte: Autor (2016).

Por este motivo o cadastro de usuários teve a inclusão do campo *checkbox* com o *label Upload* (Figura 22) que valida se o usuário pode enviar arquivos na aba Enviar Arquivo da área do *site*. Além disso, o administrador pode informar uma nova senha para o usuário, essa senha não possui validação de tamanho ou obrigatoriedade de números e letras. Não foi colocado a validação na senha para caso o administrador tenha que alterar a senha a pedido do próprio usuário, sendo esta uma senha temporária.

4.7 CONSIDERAÇÕES FINAIS

A migração de dados foi executada com sucesso. Os dados referentes a biologia molecular não foram importados do MySQL devido aos problemas encontrados nos testes com os acadêmicos Jórdan Rui da Rosa e Jovani Dalzochio, entretanto as estruturas das tabelas exportadas passaram por mudanças e testes foram realizados em conjunto antes de fazer uma importação total dos dados, garantindo a confiabilidade dos dados.

A mudança da plataforma de desenvolvimento foi um desafio proposto que proporcionou a evolução do sistema corrigindo erros, questões de performance e possibilitando a inclusão de novas funcionalidades. O IntergenicDB 2.0 é um sistema com dados consistentes e confiáveis e com uma interface moderna e de fácil acesso.

O próximo capítulo demonstrará testes efetuados na versão 2.0 contemplando a migração dos dados, o *web service*, as novas funcionalidades e o novo *layout* criado. Além disso, será explicado como publicar o portal em um *site* de hospedagem.

5 ESTUDO DE CASO

Neste capítulo serão demonstradas as novas telas de consulta da área administrativa do portal, as alterações nos cadastros existentes e os novos cadastros criados. A compatibilidade do *layout* com as outras resoluções como celulares e *tablets*.

5.1 MIGRAÇÃO DOS DADOS

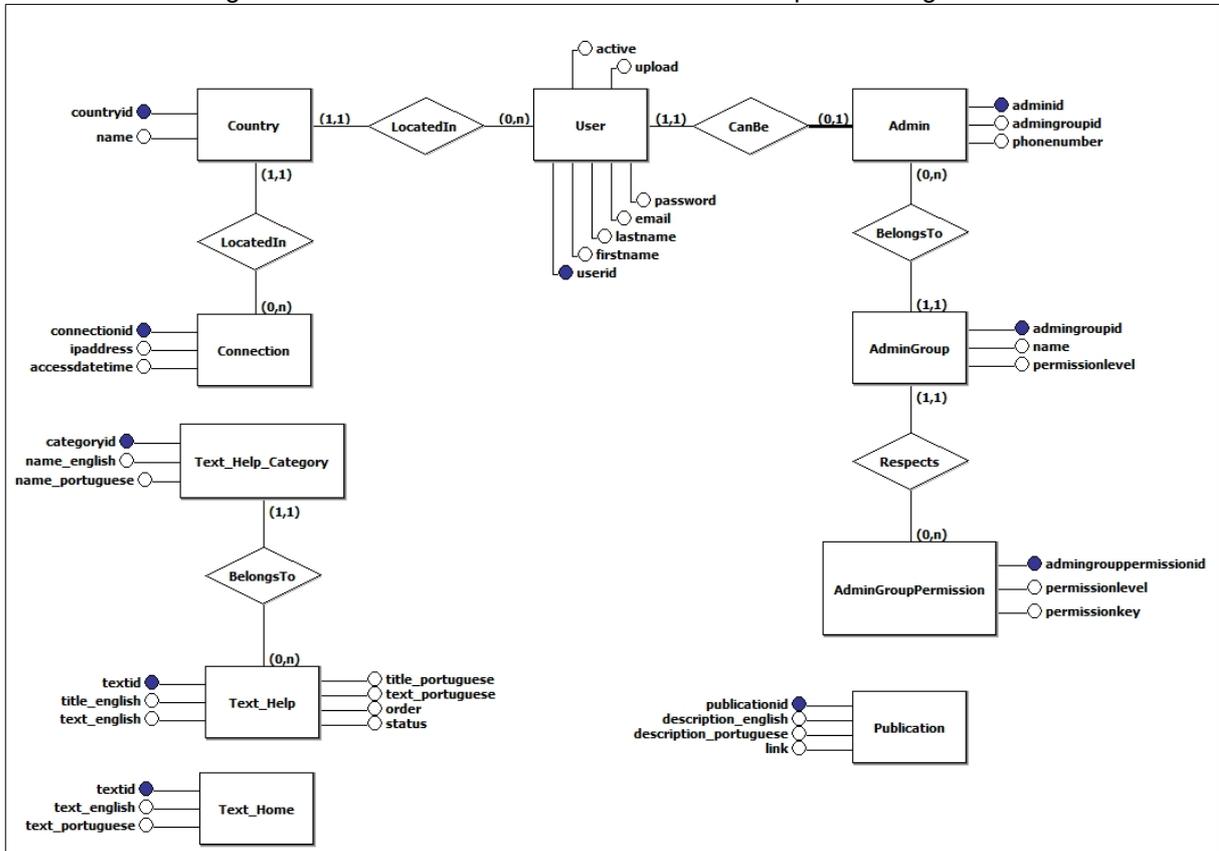
Durante a retirada do LP.Framework, percebeu-se que a forma como o *framework* de acesso aos dados foi construído afetava diretamente na performance de consulta, deixando as consultas lentas. A troca do banco de dados e a nova forma de consultar os dados, utilizando os comandos *limit* e *offset* ajudou na melhora da performance do sistema¹⁸.

Além disso, as alterações nas estruturas das tabelas e a correção na rotina de popular os dados referentes a biologia molecular também contribuíram para a performance melhorar, visto antes o banco de dados possuía um volume maior de dados que estavam incorretos. Os dados não foram testados comparando com os dados da base do MySQL devido isso, mas foram realizados testes para verificar a confiabilidade dos novos dados importados.

Devido as mudanças realizadas na estrutura de algumas tabelas durante a migração dos dados o Modelo ER da parte administrativa do portal foi atualizado conforme Figura 23. As principais alterações são as criações de 3 novas tabelas *text_help_category*, *text_help* e *text_home*, a adequação da estrutura da tabela *publication* e a exclusão das tabelas: *state* e *city*. As tabelas *useractivationpending*, *importqueue* e *importqueueentry* foram importadas, entretanto, o IntergenicDB 2.0 não está as utilizando, por este motivo não foram incluídas no modelo ER.

¹⁸ O comando OFFSET indica o início da leitura, e o LIMIT o máximo de registros a serem lidos, então para realizar a consulta dos registros de 61 a 75 utiliza-se: "SELECT coluna FROM tabela LIMIT 15 OFFSET 60".

Figura 23 – Modelo ER da área administrativa do portal IntergenicoDB



Fonte: Autor (2016).

Utilizando a ferramenta *Firebug*¹⁹ do Firefox é possível na aba *Rede* ver o tempo de demora para carregar a página. Foram realizados testes com as páginas de consultas aos dados de Gene e Gene/Organismo no portal IntergenicoDB acessando o banco de dados MySQL com o *framework* LP.Framework, onde o tempo foi de 7,04 (Figura 24) segundos e 8,02 segundos (Figura 25) respectivamente. O acesso do portal em PHP com banco de dados PostgreSQL teve um desempenho de 3,8 segundos (Figura 26) e 4,74 segundos (Erro: Origem da referência não encontrada), correspondendo uma melhor de 54% e 59% respectivamente.

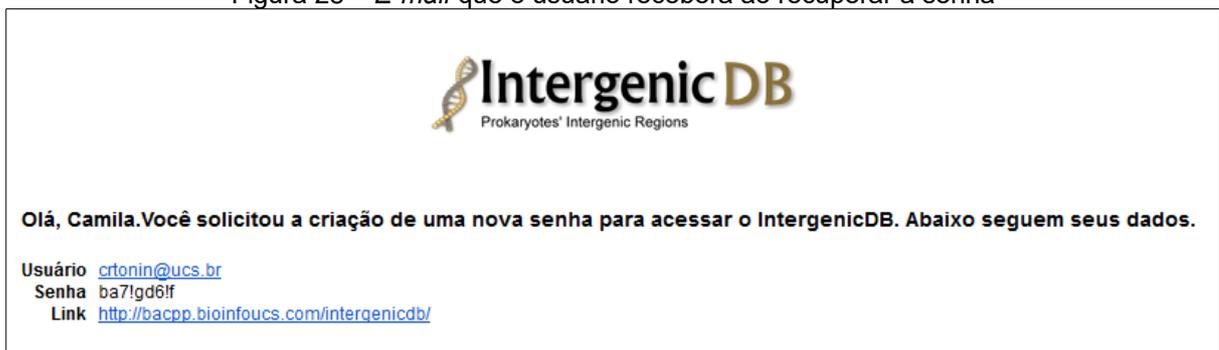
19 <https://addons.mozilla.org/pt-br/firefox/addon/firebug/>

É possível perceber nos *logs* do *Firebug* que as telas do portal IntergenicDB 2.0 possuem menos arquivos para serem carregados durante a requisição, o que também contribui para a diminuição do tempo de carregamento da página.

5.2 RECUPERAR SENHA

Ao acessar a opção de recuperar a senha e informar o *e-mail* utilizado para acessar o portal IntergenicDB será enviado um *e-mail* para o usuário conforme a Figura 28, informando uma nova senha de acesso. Após acessar o portal utilizando esta senha, se o usuário desejar, ele poderá utilizar a opção de alterar senha. A nova senha será validada e deve possuir no mínimo 6 caracteres entre letras e números.

Figura 28 – *E-mail* que o usuário receberá ao recuperar a senha



Fonte: Autor (2016).

5.3 TESTES WEB SERVICE

Para testar o *web service* foi utilizado uma extensão do navegador Chrome *Advanced REST Client*²⁰. Abaixo segue o detalhamento dos testes realizados com o *web service* implementado.

Teste	Inserir um novo usuário
Método HTTP	POST

²⁰ <https://chrome.google.com/webstore/detail/advanced-rest-client/hgmloofddffdnphfgcellkdfbfjeloo>

URL	/service/users
Dados enviados	["nome"]=> "Camila" ["sobrenome"]=>"Tonin" ["email"]=> "crtonin2@ucs.br" ["pais"]=> "Brazil" ["senha"]=> "e10adc3949ba59abbe56e057f20f883e"
Status HTTP	200: Created
Resposta	"success": "Usuário [crtonin@ucs.br] criado com sucesso!"

Teste	Inserir um novo usuário informando um e-mail que já pertence a um usuário cadastrado
Método HTTP	POST
URL	/service/users
Dados enviados	["nome"]=> "Camila" ["sobrenome"]=>"Tonin" ["email"]=> "crtonin@ucs.br" ["pais"]=> "Brazil" ["senha"]=> "e10adc3949ba59abbe56e057f20f883e"
Status HTTP	400: Bad Request
Resposta	"error": "O campo E-mail deve conter um valor único. Já existe um registro com o valor informado.\n"

Teste	Tentar inserir um novo usuário sem informar os campos usuário e senha
Método HTTP	POST
URL	/service/users
Dados enviados	["nome"]=> "Camila" ["sobrenome"]=> "Tonin" ["email"]=> "crtonin3@ucs.br"
Status HTTP	500: Internal Server Error
Resposta	"error": "\nO campo Senha é obrigatório.\nO campo País é obrigatório.\n"

Teste	Atualizando o nome do usuário inserido anteriormente
Método HTTP	PUT

URL	/service/users
Dados enviados	["nome"]=> "Camila" ["email"]=> "crtonin@ucs.br"
Status HTTP	200: OK
Resposta	"success": "Usuário [crtonin@ucs.br] atualizado com sucesso!"

Teste	Atualizando o nome do usuário com o método GET
Método HTTP	GET
URL	/service/users
Dados enviados	["nome"]=> "Camila" ["email"]=> "crtonin@ucs.br"
Status HTTP	405: Method Not Allowed

Teste	Atualizando um usuário que não existe
Método HTTP	PUT
URL	/service/users
Dados enviados	["nome"]=> "Camila" ["email"]=> "crtonin3@ucs.br"
Status HTTP	404: Not Found
Resposta	"error": "Usuário [crtonin3@ucs.br] não existe."

Teste	Validando o <i>login</i> com dados corretos
Método HTTP	GET
URL	/service/login
Dados enviados	["email"]=> "crtonin@ucs.br" ["senha"]=> "dc483e80a7a0bd9ef71d8cf973673924"
Status HTTP	302: Found
Resposta	"success": "Usuário encontrado"

Teste	Validando o <i>login</i> com dados incorretos
Método HTTP	GET

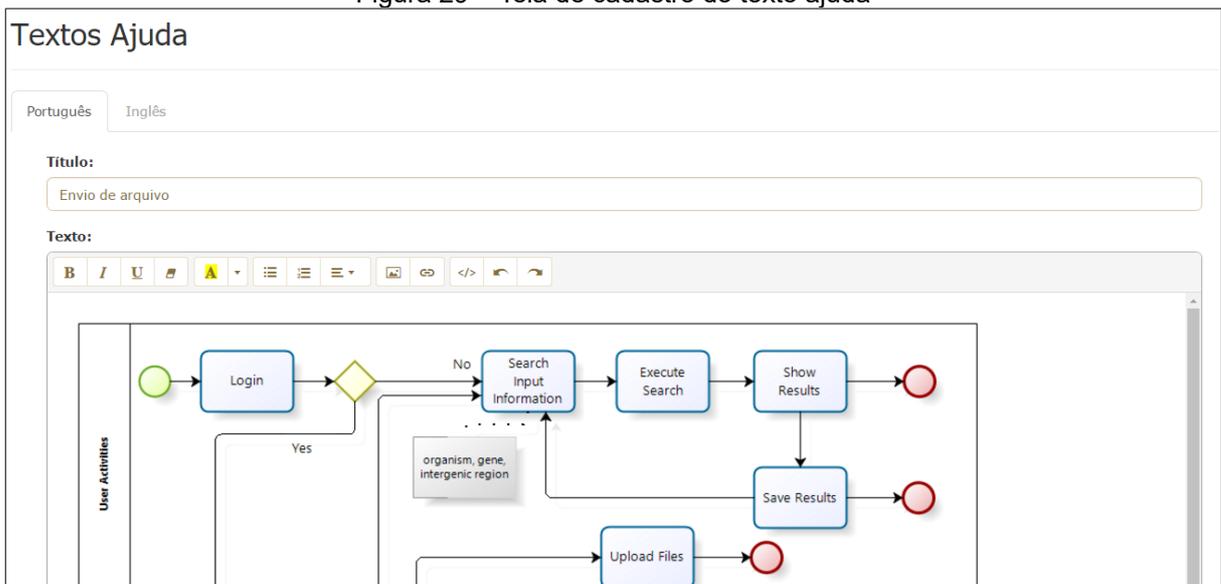
URL	/service/login
Dados enviados	["email"]=> "crtonin@ucs.br" ["senha"]=> "dc483e80a7a0bd9ef71d8cf973673924"
Status HTTP	500: Internal Server Error
Resposta	"error": "Usuário e/ou senha incorretos"

Teste	Validando o <i>login</i> utilizando o método HTTP POST
Método HTTP	POST
URL	/service/login
Dados enviados	["email"]=> "crtonin@ucs.br" ["senha"]=> "dc483e80a7a0bd9ef71d8cf973673924"
Status HTTP	405: Method Not Allowed

5.4 CADASTRO DE AJUDA

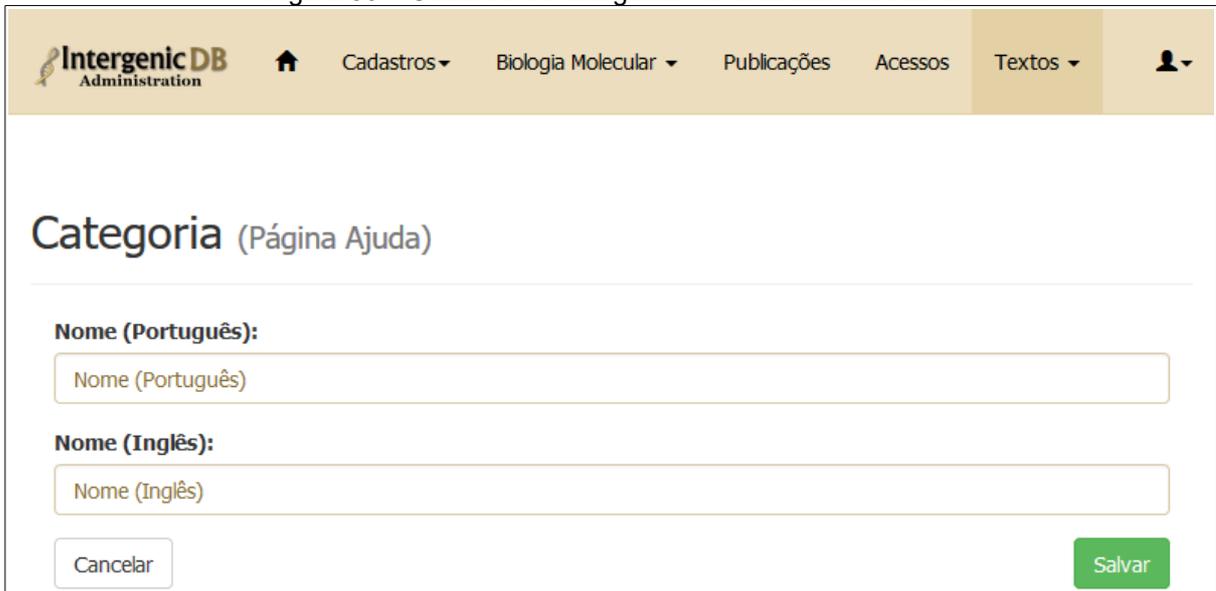
O cadastro de tela de ajuda permite a inserção de texto e imagem conforme é possível visualizar na Figura 29. É necessário informar uma categoria, que também pode ser cadastrada na área administrativa (Figura 30).

Figura 29 – Tela de cadastro do texto ajuda



Fonte: Autor (2016).

Figura 30 – Cadastro de categorias na área administrativa



The screenshot displays the 'IntergenicDB Administration' web interface. The top navigation bar includes a home icon, 'Cadastros', 'Biologia Molecular', 'Publicações', 'Acessos', 'Textos', and a user profile icon. The main content area is titled 'Categoria (Página Ajuda)'. It features two text input fields: 'Nome (Português):' and 'Nome (Inglês):'. Below these fields are two buttons: 'Cancelar' and 'Salvar'.

Fonte: Autor (2016).

5.5 NOVAS FUNCIONALIDADES

Foram implementadas novas funcionalidades no portal aproveitando a troca da linguagem e o novo *layout*.

5.5.1 Texto página inicial do *site*

O texto da página inicial (Figura 32) pode ser alterado na parte administrativa do portal pro meio da tela de cadastro, conforme Figura 31.

Figura 31 – Tela de cadastro do texto da página inicial do *site*

Texto Home

Português
Inglês

Texto:

B *I* U ~~ABC~~ A ☰ ☰ ☰ ☰ ☰ ☰ </>

Bem-vindo ao IntergenicDB!

O IntergenicDB é um banco de dados que contém sequências de DNA conhecidas como regiões intergênicas (ou promotores) de seres procaríotos.

Este banco de dados é um resultado do estudo efetuado pela Prof. Sheila de Ávila e Silva, da Universidade de Caxias do Sul (UCS), intitulado "Redes Neurais Artificiais no Reconhecimento de Regiões Promotoras em Bactérias Gram-Negativas". Este estudo investiga as regiões intergênicas, utilizando redes neurais treinadas por um programa para identificar se dada sequência de DNA é uma região promotora. Para ver mais sobre este estudo, [clique aqui](#).

O website IntergenicDB é uma ferramenta que prove uma forma simples de efetuar pesquisar e enviar dados para o banco de dados do IntergenicDB. Este website é um resultado do estudo dos alunos da Universidade de Caxias do Sul Vanessa Davanzo e Douglas Picolotto, com a orientação do Prof. Daniel Luis Notari.

[Clique aqui](#) para saber mais sobre o grupo de pesquisa de Bioinformática da Universidade de Caxias do Sul.

Cancelar
Salvar

Fonte: Autor (2016).

Figura 32 – Texto da página inicial na área do *site*



IntergenicDB

Prokaryotes' Intergenic Regions

PT | EN
Olá, Camila
[Meus dados](#) [Alterar senha](#) [Sair](#)

Início
Pesquisa
Publicações
Ajuda
Contato

Bem-vindo ao IntergenicDB!

O IntergenicDB é um banco de dados que contém sequências de DNA conhecidas como regiões intergênicas (ou promotores) de seres procaríotos.

Este banco de dados é um resultado do estudo efetuado pela Prof. Sheila de Ávila e Silva, da Universidade de Caxias do Sul (UCS), intitulado "Redes Neurais Artificiais no Reconhecimento de Regiões Promotoras em Bactérias Gram-Negativas". Este estudo investiga as regiões intergênicas, utilizando redes neurais treinadas por um programa para identificar se dada sequência de DNA é uma região promotora. Para ver mais sobre este estudo, [clique aqui](#).

O website IntergenicDB é uma ferramenta que prove uma forma simples de efetuar pesquisar e enviar dados para o banco de dados do IntergenicDB. Este website é um resultado do estudo dos alunos dos cursos de Ciência da Computação e Sistemas de Informação da Universidade de Caxias do Sul: Vanessa Davanzo, Douglas Picolotto, Jovani Dalzochio, Jórdan Rui da Rosa e Camila Rachel Tonin com a orientação do Prof. Daniel Luis Notari e da Profa. Scheila de Ávila e Silva.

[Clique aqui](#) para saber mais sobre o grupo de pesquisa de Bioinformática da Universidade de Caxias do Sul.

Fonte: Autor (2016).

5.5.2 Cadastro de publicações

As publicações podem ser informadas por meio do cadastro de publicações (Figura 33) na área administrativa do portal. A visualização das publicações ficam na aba “Publicações” na área do *site* (Figura 34).

Figura 33 – Cadastro de publicação na área administrativa do portal

Publicação

Link:

Descrição (Português):

B *I* U `</>`

NOTARI, Daniel Luis et al (Org.). **IntergenicDB: a database for intergenic sequences. Bioinformation.** Caxias do Sul, p. 381-382. Maio 2014.

Fonte: Autor (2016).

Figura 34 – Tela de publicações na área do *site*



IntergenicDB

Prokaryotes' Intergenic Regions

PT | EN

Olá, Camila

[Meus dados](#) [Alterar senha](#) [Sair](#)

Início
Pesquisa
Publicações
Ajuda
Contato

[Link] NOTARI, Daniel Luis et al. **IntergenicDB: a database for intergenic sequences. Bioinformation.** Caxias do Sul, p. 381-382. Maio 2014.

Fonte: Autor (2016).

5.5.3 Contato

Ao preencher os dados no formulário de contato e enviar é enviado um *e-mail* com as informações preenchidas pelo usuário. O *e-mail* recebido em um teste realizado pode ser visualizado na Figura 35.

Figura 35 – Exemplo de *e-mail* enviado para um novo contato realizado pelo *site*



Fonte: Autor (2016).

5.5.4 Envio de arquivo

A aba de Enviar Arquivo fica disponível apenas aos usuários que possuem permissão para fazer o *upload* de um arquivo e tal permissão é dada através do cadastro do usuário pelos administradores do portal. Os arquivos selecionados são validados podendo ser um arquivo texto ou do tipo FASTA.

Ao ser enviado um arquivo, o portal em sua versão 1.0 executava uma rotina e importava os dados utilizando uma *stored procedure*. Na versão 2.0 foi definido pelos usuários-chaves que ao ser enviado um arquivo os administradores receberão o mesmo *anexado* em um *e-mail* informando o usuário que efetuou o envio. Posteriormente, os administradores poderão analisar os dados do arquivo e importá-lo ou não. Foi efetuado um teste enviando um arquivo conforme pode ser visualizado na Figura 36.

Figura 36 – *E-mail* recebido ao ter enviado um arquivo pelo site

Fonte: Autor (2016).

5.6 LAYOUT

O novo *layout* do portal procurou manter a essência da versão 1.0 porém se propondo a incluir as novas tecnologias utilizadas como HTML 5 e a adequação as diversas resoluções existentes.

5.6.1 Área administrativa

As telas de consulta dos dados referente a biologia molecular tiveram seu *layout* alterado. No portal IntergenicDB 2.0, os dados podem apenas ser consultados, as ações de inclusão, edição e exclusão foram retiradas (Figura 37). O menu conexão foi renomeado para acessos, condizendo melhor com os dados apresentados nesta página.

Figura 37 – Tela de consulta no IntergenicDB 2.0

Gene/Organismo	Gene	Organismo	Função
SsrA-binding protein	smpB	Borrelia burgdorferi B31	Roles will be associated as soon as possible
chemotaxis protein methyltransferase	cheR	Borrelia burgdorferi B31	02030 Bacterial chemotaxis
alanine racemase	alr	Borrelia burgdorferi B31	00310 Lysine degradation

Fonte: Autor (2016).

5.6.2 Adaptável as resoluções

Figura 38 – Simulação da visualização do portal em um iPad



Fonte: Autor (2016).

O novo *layout* tanto do portal como da área administrativa foi criada de forma responsiva, adaptando-se as diversas resoluções existentes. Utilizando a ferramenta F12 do Chrome foi possível verificar como o *layout* se comportaria no *tablet* (Figura 38) e nos celulares (Figura 39).

Figura 39 – Comparação do *layout* iPhone 6 x Samsung Galaxy S5



Fonte: Autor (2016).

5.7 PUBLICAÇÃO NO SERVIDOR

Para publicar os fontes do IntergenicDB no servidor de hospedagem foi acessado o servidor via FTP Client e enviado as pastas *application*, *assets* e *system* e os arquivos *index.php*, *.htaccess* e *robots.txt*.

Foram necessários configurar alguns arquivos na pasta *application/config/*:

- *database.php* – informar os dados de acesso para o banco de dados nas respectivas variáveis: *hostname*, *username*, *password*, *database*, *schema*.
- *config.php* – configurar a variável `$config['base_url']` com a *url* de acesso, por

exemplo: <http://intergenicdb.bioinfoucs.com/>

- `email.php` – configurar os endereços de *e-mails* para enviar e receber

5.7.1 Internacionalização

A internacionalização do portal do IntergenicDB foi criada para os idiomas Português e Inglês. O *framework* CodeIgniter possui classes utilitárias que auxiliam nessa tarefa. Para isso, na pasta *application/language* foi criado as pastas “*english*” e “*portuguese*” e adicionado o arquivo *site_lang.php*, com as palavras-chaves e seus respectivos textos: `$lang['key'] = 'tradução'`²¹. Caso necessite alterar alguma informação, basta acessar o arquivo *site_lang.php* em seu respectivo idioma.

5.7.2 Arquivo .HTACCESS

O arquivo `.htaccess` é usado para retirar da url o “`index.php`” que por padrão o CodeIgniter utiliza. O arquivo deve ter as seguintes linhas, se os arquivos estiverem na raiz do servidor:

```
RewriteEngine on
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond $!^(index\.php|resources|images|robots\.txt|css|docs|js|system)
RewriteRule ^(.*)$ index.php?/$1 [L]
```

Caso os fontes sejam colocados dentro de alguma pasta específica no servidor é necessário incluir a seguinte linha ao arquivo: `RewriteBase /<pasta>`. É necessário também modificar a última linha para `RewriteRule ^(.*)$ /<pasta>/index.php?/$1 [L]`, substituindo `<pasta>` pelo caminho onde serão colocados os fontes.

²¹ Para mais informações de como utilizar a internacionalização utilizando o CodeIgniter acesse: http://www.codeigniter.com/user_guide/libraries/language.html?highlight=language

5.8 CONSIDERAÇÕES FINAIS

Todos os requisitos previstos foram desenvolvidos, alguns sofrendo modificação durante o processo, devido a alguns fatores como a retirada do LP.Framework, e outros sendo incluídos durante o desenvolvimento. O banco de dados do portal IntergenicDB continua sendo avaliado quanto a seus dados referentes as regiões intergênicas e promotores e é um banco de dados emergindo na comunidade científica em estudos sobre promotores.

As adequações que estão sendo realizadas proporcionam aos usuários confiabilidade nos dados que estão sendo acessados. A alteração do SGBD, da plataforma de desenvolvimento e do *layout* proporcionaram um ganho de performance para realizar pesquisa nos dados existentes e na navegação do portal. A versão 2.0 do portal IntergenicDB tem por objetivo torná-lo um portal conhecido e renomado dentro da comunidade científica, visto ele já possuir suas características únicas em frente aos portais existentes, como por exemplo, o tratamento único sobre as informações relacionadas a regiões intergênicas e promotores.

6 CONCLUSÃO

A questão da performance do portal ao realizar as consultas aos dados e a dificuldade em retirar o *framework* LP.Framework proporcionaram um rumo diferente na execução deste trabalho inicialmente proposta. As mudanças tornaram-se maiores em conta da troca da linguagem de programação de .NET para PHP e do novo *layout* proposto devido aos fatos do *layout* anterior: i) pouco aproveitamento da tela; ii) só estar homologado para o Internet Explorer; e iii) não possuir nenhuma compatibilidade com dispositivos móveis.

Entretanto, com a utilização do *framework* CodeIgniter, que auxilia no desenvolvimento devido ao seu objetivo de agilizar a programação e construção de projetos, o processo de desenvolvimento do IntergenicDB 2.0 pode ser realizado com sucesso no tempo proposto, incluindo a adição de novas funcionalidades: i) Página de contato na área do site; ii) Gerenciamento do texto da tela inicial; iii) Inclusão das funcionalidades alterar senha e alterar meus dados para os usuários logados na área do site e na área administrativa.

Os testes realizados em conjunto com os acadêmicos Jórdan Rui Rosa e Jovani Dalzochio também proporcionaram ganhos de performance, devido ao fato de a base anterior possuir dados inconsistentes que contribuíam para as consultas ficarem lentas. A migração dos dados foi realizada com sucesso quanto aos dados da área administrativa do portal, com dados corretos e sem perdas.

Durante o processo de migração notou-se que na tabela *users* existem usuários que se cadastraram mais de uma vez no portal, com e-mails diferentes. Na versão 1.0, o usuário não tinha permissão para efetuar login antes da ativação dos administradores. Caso essa ativação não ocorresse, os novos usuários não tinham uma forma de se comunicar com os administradores do portal.

No IntergenicDB 2.0 a funcionalidade de registrar-se permite ao usuário que efetuou o registro se logar no portal, sem a permissão de envio de arquivo. Os administradores serão informados do novo usuário e poderão efetuar a liberação

para o envio de arquivo, caso desejem. A fim de facilitar a comunicação entre os usuários e os administradores do portal o formulário de contato foi incluído no site. Sendo assim, o usuário poderá entrar em contato com os administradores solicitando a permissão para enviar arquivo caso, os administradores ainda não o tenham feito ou para enviar sugestões e melhorias.

O *web service* foi criado com o objetivo de auxiliar as aplicações externas que também utilizam a estrutura de tabelas de usuários. Com a troca do SGBD, as outras aplicações que continuam utilizando o MySQL não precisarão se conectar ao PostgreSQL, mas poderão utilizar o *web service* para manter suas funcionalidades referentes a usuários. O *framework* CodeIgniter forneceu um desenvolvimento rápido devido a sua abordagem Modelo-Visão-Controller.

A migração do código fonte do SVN para o GIT também teve alterações referentes a proposta inicial. Como houve a mudança da plataforma de desenvolvimento esta migração, na verdade, foi realizada para não se perder o que já havia sido feito do portal anteriormente e manter um histórico dos fontes da versão 1.0. Para a versão 2.0 foi criado um novo repositório no Bitbucket. O IntergenDB 2.0 propõe uma nova experiência para os usuários que acessarão a área do site e para os administradores da área administrativa, com dados consistentes e um *layout* intuitivo e de fácil visualização das informações.

6.1 TRABALHOS FUTUROS

Devido as adequações nas estruturas das tabelas referentes aos dados da biologia molecular, a importação dos arquivos enviados para os usuários não foi reimplementada na versão 2.0. Como sugestão, futuramente pode-se incluir as etapas de análise, importação e notificação do usuário na ferramenta desenvolvida para importação dos dados do acadêmico Jovani Dalzochio. Deve-se contemplar o tratamento das novas colunas e também o novo requisito de leitura diretamente do arquivo que os administradores receberam por e-mail.

REFERÊNCIAS

BROY, M. **Legacy migrationsstrategien**. Hauptseminar Management von Softwaresystemen. TU München. Fakultät für Informatik. Lehrstuhl IV: Software & Systems Engineering. München, 2005.

BRODIE, M. L.; STONEBRAKER, M. Darwin: **On the incremental migration of legacy information systems**. *Relatório Técnico TR-022-10-92-165 GTE Labs Inc*, 1993.

DALZOCHIO, Jovani. **Implementação de novas funcionalidades para o portal InterGenicDB e povoamento do seu banco de dados**. 2014. 74 f. TCC (Graduação) - Curso de Bacharelado em Ciência da Computação, Universidade de Caxias do Sul, Caxias do Sul, 2014.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de banco de dados: fundamentos e aplicações**. 6. ed. São Paulo: Addison Wesley, 2011.

FIELDING, R. T. **Architectural Styles and the Design of Network-based Software Architectures**. University of California. Irvine. 2000.

GONÇALVES, Pablo Rodrigo; SILVA DA, Ricardo Frenedoso. **Web Services – Uma Análise Comparativa**. Revista das Faculdades Integradas Claretianas. n.4, p. 7-19, dez./jan. 2012.

GUEDES, Anselmo. **Migração de dados: não deixe essa atividade tornar seu projeto um fracasso**. Disponível em: <<http://imasters.com.br/desenvolvimento/migracao-de-dados-nao-deixe-essa-atividade-tornar-seu-projeto-um-fracasso/?trace=1519021197&source=single>>. Acesso em: 11 de mar. 2016.

HEUSER, Carlos Alberto. **Projeto de banco de dados**. 6. ed. Porto Alegre: Bookman, 2009.

IGAWA, R. A.. **Uma proposta de reengenharia com abordagem incremental para sistemas orientado a objeto**. 2013. 77 f. TCC (Graduação). Bacharelado em Ciência da Computação – Universidade Estadual de Londrina, Londrina, 2013.

ISO/IEC. **Systems and software engineering — software life cycle processes**. [S.l.], 1998.

MALACRIDA, Tamiris Fernanda et al. **Migração de dados entre bancos de dados relacionais**. Colloquium Exactarum, v. 6, n.1, Jan-Jun. 2014, p.20 – 36. DOI: 10.5747/ce.2014.v06.n1.e066

MASON, M. **Pragmatic Version Control: Using Subversion (The Pragmatic Starter Kit Series)**. Pragmatic Bookshelf, 2006.

MENDONÇA, Manoel H. R. de. **Metodologia de Migração de dados em um contexto de migração de sistemas legados**. 2009 151 f. Dissertação (Pós-Graduação em Ciência da Computação). Universidade Federal de Pernambuco, Pernambuco, 2009.

MYSQL. Disponível em: <https://dev.mysql.com/doc/mysql-reslimits-excerpt/5.6/en/>
Acesso em: 2 de mar. 2016

NOTARI, Daniel Luis et al (Org.). **IntergenicDB: a database for intergenic sequences**. *Bioinformatics*. Caxias do Sul, p. 381-382. Maio 2014. Disponível em: <<http://www.bioinformatics.net/010/97320630010381.pdf>>. Acesso em: 02 set. 2015.

_____. **Desenvolvimento de workflows científicos para a geração e análise de diferentes redes de interatomos**. 2012. xix, 161 f. Tese (Doutorado) - Universidade de Caxias do Sul, Programa de Pós-Graduação em Biotecnologia, 2012.

OLIVEIRA, C. S; MARCELINO M. A. **Metodologias e estratégias de migração de dados**. v. 13, n. 3, p. 183-191. São Paulo: Sinergia, 2012

PHPMyAdmin. Disponível em: <https://www.phpmyadmin.net/about/> Acesso em: 2 de mar. 2016

PRESSMAN, R., MAXIM, B. **Engenharia de software**. 8. ed. Porto Alegre: AMGH, 2016.

PRESSMAN, Roger S.. **Engenharia de Software: Uma Abordagem Profissional**. 7. ed. Porto Alegre: AMGH, 2011.

RICHARDSON, L.; RUBY, S. **RESTful Web Services**. [S.l.]: O'Reilly, 2007.

ROB, P.; CORONEL, C. **Sistemas de banco de dados – Projeto, implementação e administração**. São Paulo: Editora CENGAGE Learning, 2011.

RODRIGUES-NETO, J.; PASSOS, E. **JExodus: uma ferramenta de migração de dados independente de SGBD**. Escola Regional de Computação Ceará, Maranhão, Piauí (ERCEMAPI), Parnaíba, PI, 2009.

ROSA, Jórdan Rui. **Criação e implantação de um data warehouse para reformatar o mecanismo de pesquisa do portal IntergenicDB 2.0**. 2016. 130 f. TCC (Graduação) - Curso de Bacharelado em Ciência da Computação, Universidade de Caxias do Sul, Caxias do Sul, 2016.

SUEHRING, Steve. **MySQL: Bible**. New York. Wiley Publishing, Inc, 2002.

TRIPATHY, P.; NAIK, K. **Taxonomy of software maintenance and evolution, in software evolution and maintenance: a practitioner's approach**. Hoboken: John Wiley & Sons, Inc., 2015.

SOMMERVILLE, I. **Engenharia de software**. 9. ed. São Paulo: Addison Wesley, 2011.

WU, Bing et al. **Legacy systems migration – a method and its Tool-Kit framework**. Asia Pacific Software Engineering Conference e International Computer Science Conference. Páginas: 312 – 320, Dezembro 1997.

APÊNDICE A – PORTAL INTERGENIGDB

O portal IntergenicDB (Figura 40) é um repositório público no qual pesquisadores podem ter acesso aos dados do banco de dados PromotoresDB – que armazena às informações de sequências de nucleotídeos de regiões promotoras – desenvolvido para ser uma ferramenta de consulta e para o *download* e *upload* de arquivos (PICOLOTTO, 2012; DALZOCHIO, 2014; NOTARI et al., 2014). Ambos foram criados para atender as necessidades do grupo de Bioinformática da Universidade de Caxias do Sul.

Figura 40 – Página inicial do portal IntergenicDB



IntergenicDB
Prokaryotes' Intergenic Regions

Início | Pesquisar | Publicações | Ajuda

Bem-vindo ao IntergenicDB!

O IntergenicDB é um banco de dados que contém sequências de DNA conhecidas como regiões intergênicas (ou promotores) de seres procariotos.

Este banco de dados é um resultado do estudo efetuado pela Prof. Sheila de Ávila e Silva, da [Universidade de Caxias do Sul \(UCS\)](#), intitulado "Redes Neurais Artificiais no Reconhecimento de Regiões Promotoras em Bactérias Gram-Negativas". Este estudo investiga as regiões intergênicas, utilizando redes neurais treinadas por um programa para identificar se dada sequência de DNA é uma região promotora. Para ver mais sobre este estudo, [clique aqui](#).

O website IntergenicDB é uma ferramenta que prove uma forma simples de efetuar pesquisar e enviar dados para o banco de dados do IntergenicDB. Este website é um resultado do estudo dos alunos da Universidade de Caxias do Sul Vanessa Davanzo e Douglas Picolotto, com a orientação do Prof. Daniel Luis Notari.

[Clique aqui](#) para saber mais sobre o grupo de pesquisa de Bioinformática da Universidade de Caxias do Sul.

Entrar

Usuário:

Senha:

[Ainda não está registrado?](#)
[Esqueceu sua senha?](#)

Fonte: IntergenicDB²² (2016)

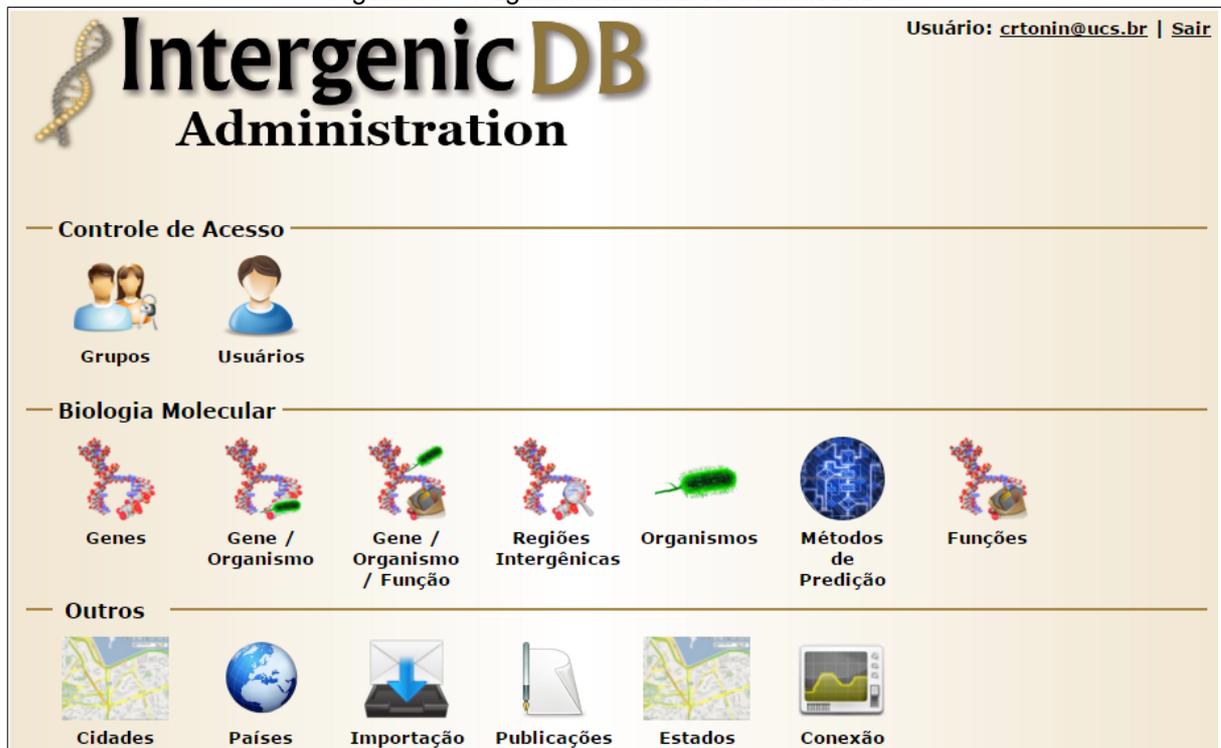
O portal possui acesso público, permitindo que as consultas sejam livremente utilizadas por qualquer usuário. O resultado da consulta para um usuário não registrado somente é disponibilizado em tela. Para o download da informação é

²² Disponível em: <<http://intergenicdb.bioinfocps.com>>. Acesso em: 01/04/2016.

necessário possuir usuário e senha (DAVANZO, 2010; PICOLOTTO, 2012; DALZOCHIO, 2014).

A área administrativa do portal (Figura 41), principal foco deste trabalho, foi desenvolvida pelo aluno Douglas Picolotto (PICOLOTTO, 2012), e seu acesso é realizado apenas por usuários que estejam devidamente cadastrados e autorizados. Para que um novo utilizador tenha acesso é necessário a realização do cadastro (Figura 6) e que seja autorizado por um dos membros com tal privilégio. (ROSA, 2015)

Figura 41 – Página inicial da área administrativa



Fonte: IntergenicDB²³ (2016).

Na Figura 41, percebe-se que itens estão separados em três grupos: Controle de Acesso, Biologia Molecular e Outros. Segundo ROSA (2015) podem ser definidos:

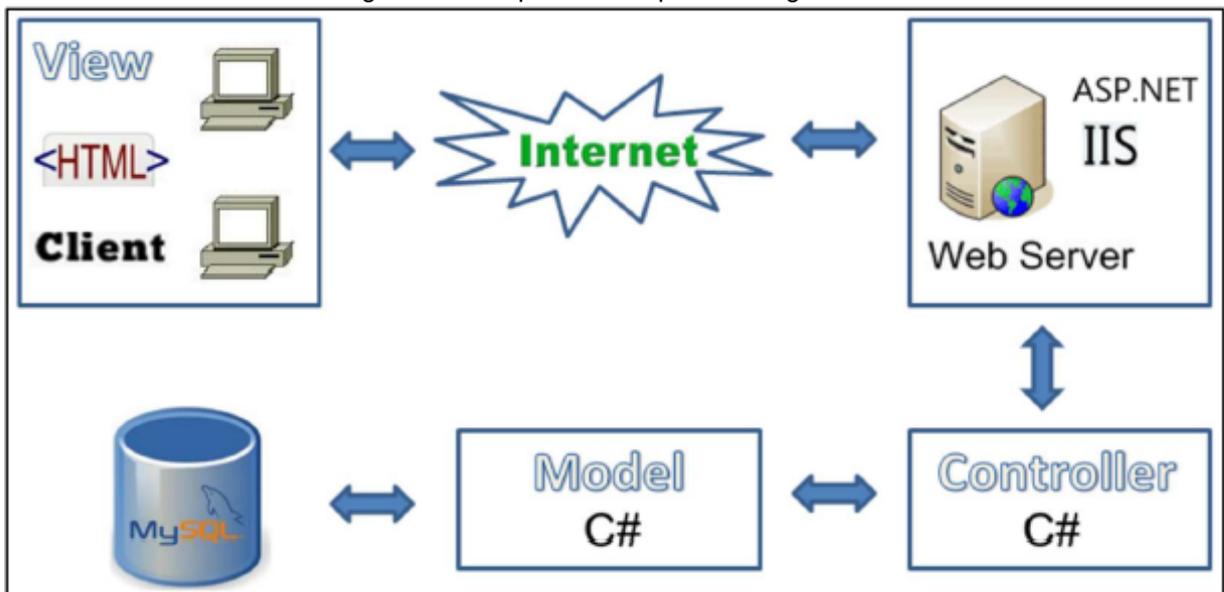
²³ Disponível em: <<http://intergenicdb.bioinfocps.com/admin>>. Acesso em: 01/04/2016.

- **Controle de Acesso:** gerencia de usuário e grupos do portal. O grupo ao qual o usuário está associado lhe proverá mais ou menos acessos.
- **Biologia Molecular:** gerencia os dados biológicos inseridos no banco de dados. Apesar de a ferramenta permitir a inserção, seu principal objetivo é a manutenção (PICOLOTTO, 2012).
- **Outros:** é o mais heterogêneo e voltado a dados de geoposição dos usuários que utilizam o portal, utilizando-os para cálculos e estatísticas.

O item Importação, é uma ferramenta que não importa dados externos no banco de dados, apesar do nome, mas é utilizada pelos administradores para validar as novas informações que foram importadas para o banco de dados. Somente após a validação do administrador, alguma informação nova será visível na tela de consulta. Caso contrário, ela pode ser excluída do banco (ROSA, 2015).

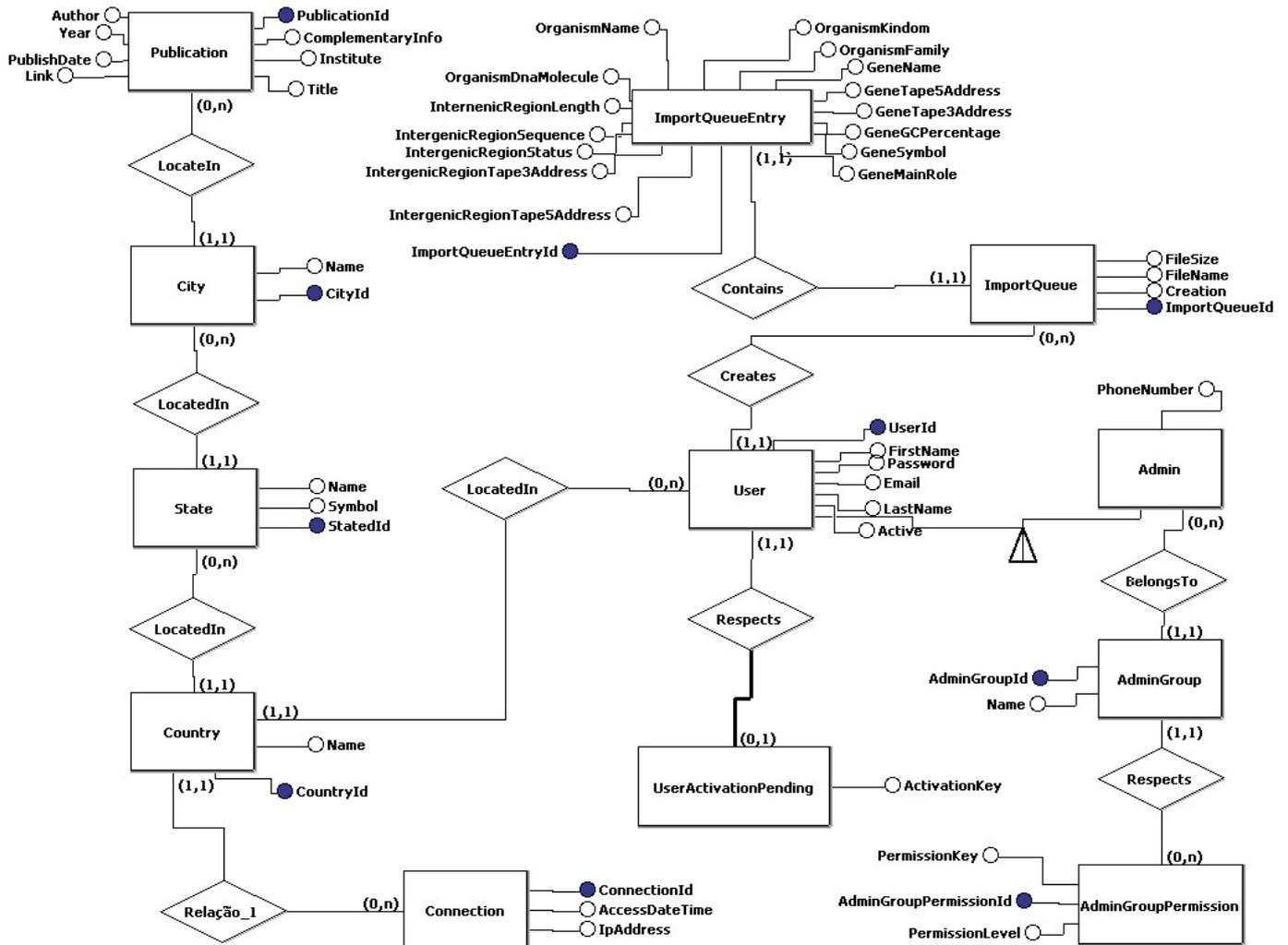
O portal está armazenado em um provedor que comporta os dois SGBD PostgreSQL e MySQL e o WebServer IIS. O portal foi criado utilizando o SGBD MySQL e a plataforma de desenvolvimento web da Microsoft, a ASP.NET com a linguagem de programação C Sharp (C#) e padrão de arquitetura MVC. A arquitetura está representada na Figura 42 (DAVANZO, 2010; DALZUCHIO, 2014).

Figura 42 – Arquitetura do portal IntergenicDB



Fonte: Dalzochio (2010, p. 27).

ANEXO A – MODELO ER ÁREA ADMINISTRATIVA DO PORTAL



Fonte: IntergenicDB²⁴ (2016).

24 Disponível em: <http://intergenicdb.bioinfocms.com/Content/images/portal_intergenic_relation_model.jpg>. Acesso em: 19 abr 2016.