

**UNIVERSIDADE DE CAXIAS DO SUL**  
**Centro de Computação e Tecnologia da Informação**  
**Curso de Bacharelado em Sistemas de Informação**

**DOUGLAS PICOLOTTO**

**IMPLEMENTAÇÃO DE NOVAS FUNCIONALIDADES PARA O PORTAL  
INTERGENICDB**

**CAXIAS DO SUL**  
**2012**

**UNIVERSIDADE DE CAXIAS DO SUL**  
**Centro de Computação e Tecnologia da Informação**  
**Curso de Bacharelado em Sistemas de Informação**

**DOUGLAS PICOLOTTO**

**IMPLEMENTAÇÃO DE NOVAS FUNCIONALIDADES PARA O PORTAL  
INTERGENICDB**

Trabalho de Conclusão de Curso do Título de  
Bacharel pela Universidade de Caxias do Sul.  
Área de concentração: Biologia Molecular  
Orientador Prof. Ms. Daniel Luis Notari

**CAXIAS DO SUL**  
**2012**

## RESUMO

Com as tecnologias existentes atualmente, principalmente na área de Informática, é possível armazenar e processar dados sobre Biologia Molecular de forma rápida e consistente. Esta união entre Biologia Molecular e Informática originou uma nova ciência chamada Bioinformática. Atualmente existem vários portais que tratam e expõem estas informações auxiliando diversas outras ciências como a Medicina, Agropecuária e a Indústria Farmacêutica, na identificação de doenças, criação de medicamentos, entre outros. Contudo, estes portais não atendiam as necessidades do grupo de Bioinformática da Universidade de Caxias do Sul, e diante disto o Aluno Aurione Francisco Molin e a aluna Vanessa Davanzo desenvolveram, em seus trabalhos de conclusão de curso, o portal IntergenicDB seguindo as orientações da profa. Dra. Scheila de Ávila e Silva. O principal propósito do portal é armazenar e disponibilizar informações sobre regiões intergênicas e métodos de predição de promotores. Contudo, era necessário que novos requisitos fossem atendidos para disponibilizá-lo a comunidade científica. Estes requisitos foram definidos em conjunto com a profa. Dra. Scheila de Ávila e Silva, e com base neles foi criado um projeto que possuía os detalhes e os artefatos necessários para a solução dos mesmos. Depois de desenvolvida a solução, testes de consulta foram efetuados para assegurar o bom funcionamento da principal funcionalidade do portal. Ao final do processo, o IntergenicDB tornou-se um portal mais robusto, mais fácil de utilizar e com um maior controle sobre as informações.

**Palavras-Chave:** IntergenicDB. Biologia Molecular. Bioinformática. Regiões Intergênicas. Promotores.

## ABSTRACT

With the today existing technologies, mainly in IT, it's possible store and process Molecular Biology data in a quick and consistent manner. This union of Molecular Biology and Informatics created a new science called Bioinformatics. Currently there are several portals that process and expose this kind of information helping several other sciences such as Medicine, Agriculture and the Pharmaceutical Industry in the identification of diseases, creation of new drugs, among others. However, these portals did not meet the needs of the Bioinformatics group at the University of Caxias do Sul, in this situation the students Aurione Francisco Molin and Vanessa Davanzo developed, in their final papers, the portal IntergenicDB following the guidance of Profa. Dra. Scheila de Ávila e Silva. The main purpose of the portal is to store and provide information on intergenic regions and promoter prediction methods. However, it was necessary that new requirements were met to make it available to the scientific community. These requirements were defined together with Profa. Dra. Scheila de Ávila e Silva, and based on them was created a project that had all the details and artifacts needed for their solution. Once developed the solution, tests over the search engine were performed to ensure the smooth operation of the main function of the portal. At the end of the process, IntergenicDB has become a more robust portal, easier to use and with greater control over the informations.

**Keywords:** IntergenicDB. Molecular Biology. Bioinformatics. Intergenic Regions. Promoters.

## LISTA DE ILUSTRAÇÕES

Figura 1: Diferenças entre células procarióticas e eucarióticas .....	13
Figura 2: Estrutura do DNA .....	15
Figura 3: Dogma Central da Biologia Molecular .....	16
Figura 4: Processo de replicação de DNA .....	17
Figura 5: Processo de transcrição de DNA .....	18
Figura 6: Processo de tradução do mRNA.....	19
Figura 7: Promotor .....	20
Figura 8: Portal NCBI .....	23
Figura 9: Portal ENA .....	25
Figura 10: Portal DDBJ .....	27
Figura 11: Portal RegulonDB.....	28
Figura 12: Portal JCVI-CMR.....	29
Figura 13: Modelo Conceitual do protótipo PromotoresDB .....	33
Figura 14: Portal IntergenicDB .....	34
Figura 15: Modelo Conceitual do Banco de Dados .....	34
Figura 16: Arquitetura do sistema .....	36
Figura 17: Diagrama de Classes .....	37
Figura 18: Modelo de Navegação.....	38
Figura 19: Tela de busca do IntergenicDB .....	38
Figura 20: Casos de uso das novas funcionalidades .....	40
Figura 21: <i>Login</i> de Administração .....	41
Figura 22: <i>Home</i> de Administração.....	41
Figura 23: Modelo de navegação de telas da área de administração.....	42
Figura 24: Exemplo de tela de manutenção .....	42
Figura 25: Exemplo de tela de cadastro (Inclusão) .....	43
Figura 26: Cadastro de usuários do tipo <i>user</i> .....	44
Figura 27: Cadastro de grupos de usuários .....	45
Figura 28: Cadastro de usuário básico ( <i>user</i> ).....	45
Figura 29: Cadastro de usuário avançado ( <i>admin</i> ) .....	46
Figura 30: <i>Upload</i> de arquivos .....	46
Figura 31: Manutenção da fila de importação .....	47
Figura 32: Área de postagem .....	49
Figura 33: Tela de pesquisa.....	50
Figura 34: Modelo conceitual IntergenicDB 2.0.....	51
Figura 35: Modelo lógico IntergenicDB 2.0 .....	53
Figura 36: Diagrama de classes do requisito 1 .....	54
Figura 37: Diagrama de classes do requisito 2 .....	55
Figura 38: Diagrama de classes do requisito 3 .....	56
Figura 39: Diagrama de classes do requisito 5 .....	57
Figura 40: Diagrama de classes do requisito 6 .....	58
Figura 41: Diagrama de classes do requisito 7 .....	59
Figura 42: <i>Microsoft Visual Studio 2010</i> .....	60
Figura 43: <i>Paint .NET</i> .....	60

Figura 44: MySQL <i>Query Browser</i> .....	61
Figura 45: Detalhes de implementação no <i>AccountController</i> .....	62
Figura 46: Arquivo de configuração dos menus.....	63
Figura 47: Detalhes da classe de serviço <i>MenuProvider</i> .....	64
Figura 48: Detalhes de implementação do <i>MenuProvider</i> .....	65
Figura 49: Detalhes de implementação da operação <i>Maintenance</i> .....	65
Figura 50: Carga dos dados da janela de seleção do <i>picklist (javascript)</i> .....	66
Figura 51: Carga dos dados da janela de seleção do <i>picklist (AdministrationController)</i> .....	67
Figura 52: Configuração SMTP no <i>web.config</i> .....	68
Figura 53: Detalhes de implementação da operação <i>ArrangeData</i> .....	69
Figura 54: Formato do arquivo do <i>Prompredict</i> .....	70
Figura 55: Detalhes de implementação da aba <i>News</i> .....	71
Figura 56: Detalhes da utilização do paginador na aba <i>News</i> .....	71
Figura 57: Arquivos de <i>Resources</i> .....	72
Figura 58: Detalhes do serviço <i>GlobalizationHelper</i> .....	72
Figura 59: Tela de resultados com paginação.....	73
Figura 60: Detalhes de implementação da aba <i>Search</i> .....	73
Figura 61: Seção do resultado de pesquisa por nome do organismo .....	76
Figura 62: Resultado de pesquisa por nome do gene.....	77
Figura 63: Seção do resultado de pesquisa por .....	77
Figura 64: Exemplo de resultado com método de predição .....	78
Figura 65: Seção do resultado da pesquisa organismo x gene.....	79
Figura 66: Seção do resultado da pesquisa gene x região intergênica.....	79
Figura 67: Seção do resultado da pesquisa gene x método de predição.....	80
Figura 68: <i>Wizard</i> de instalação do <i>MySQL Community Server</i> .....	91
Figura 69: Janela de serviços do <i>windows (Windows Services)</i> .....	92
Figura 70: Instalação do <i>MySQL GUI Tools</i> .....	92
Figura 71: <i>MySQL Query Browser</i> .....	93
Figura 72: Pools de aplicativos do IIS 8 .....	94
Figura 73: Publicação no <i>Microsoft Visual Studio</i> .....	95
Figura 74: Configuração da publicação .....	95
Figura 75: Tela da opção <i>Connection</i> .....	96
Figura 76: Converter aplicativo no IIS .....	97
Figura 77: Configurações de Adição de Aplicativo.....	97
Figura 78: Configuração da conexão com o banco de dados .....	98
Figura 79: Teste da instalação.....	99

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	10
1.1	PROBLEMA DE PESQUISA.....	10
1.2	QUESTÃO DE PESQUISA.....	11
1.3	OBJETIVOS .....	11
1.3.1	<b>Objetivo Geral</b> .....	11
1.3.2	<b>Objetivos Específicos</b> .....	11
1.4	ESTRUTURA DO TEXTO .....	11
<b>2</b>	<b>BIOLOGIA MOLECULAR</b> .....	13
2.1	A CÉLULA .....	13
2.2	ORGANISMOS PROCARIONTES (PROCARIOTOS).....	14
2.3	DNA E RNA .....	14
2.4	DOGMA CENTRAL DA BIOLOGIA MOLECULAR.....	16
2.4.1	<b>Replicação</b> .....	16
2.4.2	<b>Transcrição</b> .....	17
2.4.3	<b>Tradução</b> .....	19
2.5	PROMOTORES .....	20
2.6	CONSIDERAÇÕES FINAIS .....	21
<b>3</b>	<b>BIOINFORMÁTICA</b> .....	22
3.1	BANCOS DE DADOS DE BIOLOGIA MOLECULAR .....	22
3.2	PORTAIS .....	23
3.2.1	<b>NCBI</b> .....	23
3.2.2	<b>ENA</b> .....	25
3.2.3	<b>DDBJ</b> .....	26
3.2.4	<b>RegulonDB</b> .....	28
3.2.5	<b>JCVI-CMR</b> .....	29
3.3	CONSIDERAÇÕES FINAIS .....	30
<b>4</b>	<b>PORTAL INTERGENICDB</b> .....	32
4.1	PROMOTORESDB .....	32
4.2	INTERGENICDB .....	33
4.3	CONSIDERAÇÕES FINAIS .....	39
<b>5</b>	<b>INTERGENICDB V2.0</b> .....	40

5.1	REQUISITOS .....	40
5.1.1	<b>Manutenção dos dados</b> .....	40
5.1.2	<b>Autenticação e autorização de usuários</b> .....	44
5.1.3	<b>Upload de arquivos</b> .....	46
5.1.4	<b>Carga de dados</b> .....	48
5.1.5	<b>Área de postagem</b> .....	49
5.1.6	<b>Internacionalização</b> .....	49
5.1.7	<b>Consultas e testes</b> .....	50
5.2	MODELO CONCEITUAL (ER) .....	51
5.3	DIAGRAMAS DE CLASSES .....	54
5.4	FERRAMENTAS.....	59
5.4.1	<b>Visual Studio 2010</b> .....	59
5.4.2	<b>Paint .NET</b> .....	60
5.4.3	<b>MySQL Query Browser</b> .....	61
5.5	CONSIDERAÇÕES FINAIS .....	61
<b>6</b>	<b>DESENVOLVIMENTO</b> .....	<b>62</b>
6.1	MANUTENÇÃO DOS DADOS.....	62
6.1.1	<b>Login</b> .....	62
6.1.2	<b>Administração</b> .....	63
6.1.3	<b>Manutenção</b> .....	65
6.1.4	<b>Cadastros</b> .....	66
6.2	AUTENTICAÇÃO E AUTORIZAÇÃO DE USUÁRIOS .....	67
6.3	UPLOAD DE ARQUIVOS.....	68
6.4	CARGA DOS DADOS .....	70
6.5	ÁREA DE POSTAGEM .....	71
6.6	INTERNACIONALIZAÇÃO .....	71
6.7	CONSULTAS E TESTES.....	72
6.8	DIFICULDADES ENCONTRADAS .....	73
6.9	CONSIDERAÇÕES FINAIS .....	74
<b>7</b>	<b>ESTUDO DE CASO</b> .....	<b>75</b>
7.1	AMBIENTE DE TESTES.....	75
7.2	CONSULTAS SIMPLES .....	75
7.2.1	<b>Organismo</b> .....	75
7.2.2	<b>Gene</b> .....	76



7.2.3	Região Intergênica.....	77
7.2.4	Método de Predição.....	78
7.3	CONSULTAS COMBINADAS .....	78
7.3.1	Organismo x Gene.....	78
7.3.2	Gene x Região Intergênica.....	79
7.3.3	Gene x Método de Predição.....	80
7.4	CONSIDERAÇÕES FINAIS .....	80
8	CONCLUSÃO .....	82
8.1	TRABALHOS FUTUROS.....	83
9	REFERÊNCIAS .....	84
	ANEXO A – Diagrama de classes do projeto <i>LP.Framework</i> .....	86
	ANEXO B – Exemplo de arquivo de upload do IntergenicDB .....	87
	ANEXO C – Diagrama de classes atualizado do projeto IntergenicDB.Data .....	88
	ANEXO D – Diagrama de classes atualizado do projeto IntergenicDB.Business .....	89
	ANEXO E – Diagrama de classes atualizado do projeto IntergenicDB .....	89
	ANEXO F – Processo de instalação do IntergenicDB.....	91
	<b>Instalação do <i>MySQL</i></b> .....	91
	<b>Instalação da aplicação</b> .....	94

## 1 INTRODUÇÃO

A biologia molecular é uma ciência dentro da área de biologia que estuda os processos internos da célula, mais precisamente o material genético. Seu estudo e compreensão têm auxiliado as mais diversas áreas de conhecimento, como a biologia, a medicina, a agropecuária, entre outras. (ZAHA, 1996).

Dentre as muitas áreas da biologia molecular, pode-se incluir o estudo do material genético de cada ser vivo, composto por moléculas chamadas DNA. Estas moléculas são formas helicoidais constituídas por duas fitas antiparalelas de estruturas chamadas nucleotídeos. O DNA pode ser dividido em seções denominadas genes e cada seção possui funções específicas dentro da célula (ALBERTS *et al.*, 1999; GRIFFITHS *et al.*, 2002; ZAHA, 1996).

Um gene pode ser dividido em duas partes: uma região não-codificante, conhecida como região intergênica, onde estão localizadas as informações sobre a regulação do processo de transcrição do gene, e a região codificante que possui a sequência que será transcrita. Nem todos os genes de uma cadeia de DNA são expressos durante o processo de transcrição, sendo que existem sequências na região intergênica que indicam se o gene será expresso. Estas são chamadas de sequências promotoras, ou simplesmente promotores (ALBERTS *et al.*, 2010; ZAHA, 1996).

A quantidade de dados biológicos disponíveis atualmente é muito grande, tornando inviável o seu processamento e análise por seres humanos. Contudo, graças aos avanços tecnológicos foi possível utilizar a informática para efetuar este processamento e análise, que por sinal, trouxe outros benefícios como a possibilidade de efetuar cálculos matemáticos complexos e armazenar dados gerados de forma eficiente em bancos de dados de biologia molecular. Esta necessidade de envolver a Biologia Molecular e a Informática criou uma nova ciência chamada Bioinformática (LESK, 2008).

### 1.1 PROBLEMA DE PESQUISA

Atualmente existem diversos portais *web* que disponibilizam informações sobre Biologia Molecular. Contudo o grupo de bioinformática da Universidade de Caxias do Sul possui necessidades específicas quanto ao armazenamento e processamento destas informações. Diante disto, o aluno Aurione Francisco Molin (MOLIN, 2009) definiu um protótipo de uma base de dados chamado PromotoresDB em seu trabalho de conclusão de curso, que atendesse as necessidades do grupo de bioinformática da UCS para armazenamento

e consultas intergênicas auxiliando na predição de promotores. A aluna Vanessa Davanzo (DAVANZO, 2010) também em seu trabalho de conclusão, desenvolveu um portal para consulta de sequências intergênicas chamado IntergenicDB, utilizando como base o protótipo de Molin. O portal desenvolvido permite efetuar consultas de sequências intergênicas com possível função biológica de DNA de organismos procariontes (Davanzo, 2010).

## 1.2 QUESTÃO DE PESQUISA

Quais as tecnologias e modificações devem ser definidas para transformar o IntergenicDB em um portal de consulta de regiões promotoras?

## 1.3 OBJETIVOS

### 1.3.1 Objetivo Geral

O objetivo deste trabalho é transformar e habilitar o IntergenicDB para ser um portal de consultas para regiões promotoras.

### 1.3.2 Objetivos Específicos

- Definir os novos requisitos do sistema;
- Selecionar as ferramentas, artefatos a serem utilizadas para o projeto e desenvolvimento do sistema;
- Criação do projeto das novas funcionalidades, com os padrões de arquitetura a serem utilizados;
- Implementação do sistema;
- Criação e execução de Testes;
- Avaliação dos resultados obtidos;

## 1.4 ESTRUTURA DO TEXTO

O capítulo 2 apresenta um breve resumo sobre Biologia Molecular para conceituar o conteúdo que será visto nos demais capítulos, abordando de forma sucinta a estrutura celular de organismos procariontes, material genético (DNA o RNA), os processos intracelulares (Dogma Central da Biologia Molecular) e promotores.

O capítulo 3 conceitua os termos Bioinformática e Bancos de Dados de Biologia Molecular, além disto mostra alguns portais *web* existentes que manipulam e disponibilizam dados sobre Biologia Molecular, para exemplificar e comparar com o portal desenvolvido neste trabalho, o IntergenicDB.

O capítulo 4 apresenta um histórico do desenvolvimento do IntergenicDB, desde a concepção do protótipo do banco de dados chamado PromotoresDB (MOLIN, 2009), até a criação do portal IntergenicDB (DAVANZO, 2010).

O capítulo 5 expõe os novos requisitos que são indispensáveis para permitir a utilização do IntergenicDB em produção. São definidas também as soluções para os mesmos, os artefatos necessários para o desenvolvimento e as ferramentas que serão utilizadas durante o processo.

O capítulo 6 demonstra o processo de desenvolvimento da nova versão do IntergenicDB, elucidando alguns dos principais detalhes do código fonte escrito, e algumas das principais dificuldades encontradas.

O capítulo 7 mostra a execução das consultas definidas nos requisitos expostos no capítulo 5, apresentando os resultados do desempenho desta execução e também as telas do IntergenicDB resultantes das consultas.

O capítulo 8 traz a conclusão deste trabalho e algumas das sugestões para trabalhos futuros.

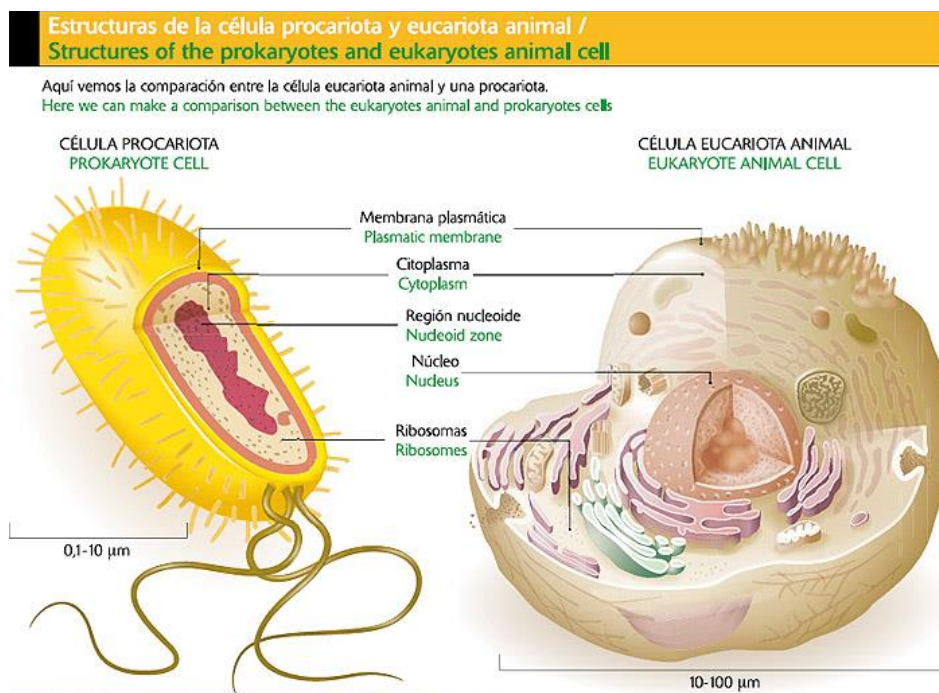
## 2 BIOLOGIA MOLECULAR

Biologia Molecular é a ciência que estuda os processos intracelulares relacionados ao DNA e seus constituintes. Neste capítulo, serão apresentados os conceitos básicos desta ciência, para possibilitar o entendimento das demais seções deste trabalho.

### 2.1 A CÉLULA

Constituída basicamente por membranas e por uma solução aquosa, a célula é considerada a menor unidade de vida pela moderna teoria celular. Tendo uma capacidade extraordinária de replicação, todos os seres vivos são compostos por uma infinidade de células derivadas de uma única célula fundadora (ZAHA, 1996; ALBERTS *et al.*, 1999).

Mesmo com uma série de similaridades, os seres vivos podem ser classificados em dois grandes grupos de acordo com sua estrutura celular (Figura 1) (ZAHA, 1996):



**Figura 1: Diferenças entre células procarióticas e eucarióticas**

**Fonte: C.B.C<sup>1</sup>**

- Os Procariotos, que não possuem um envoltório nuclear em suas células. Neste grupo de organismos estão as bactérias, as arqueobactérias e as algas azuis;
- Os Eucariotos, que possuem um envoltório nuclear. Neste grupo de organismos estão todos os organismos multicelulares e também alguns unicelulares.

<sup>1</sup> <http://biologia-cbc.es.tl/Virus,-c-e2-lulas-procariontes-y-eucariontes.htm#>

Na seção 2.2 serão detalhados os procariotos, foco deste trabalho.

## 2.2 ORGANISMOS PROCARIONTES (PROCARIOTOS)

Os procariotos, por possuírem uma estrutura mais simples, tem um tamanho menor do que os eucariotos medindo poucos micrômetros de comprimento, e geralmente possuem uma forma arredondada ou de bastonetes. Não possuem um invólucro nuclear em suas células, contudo existem regiões no interior da mesma chamadas nucleóides. O nucleóide é semelhante a uma massa arredondada, formado por um filamento circular de DNA<sup>2</sup>, e localizado próximo ou mesmo ligado à membrana plasmática (ALBERTS *et al.*, 2010; JUNQUEIRA E CARNEIRO, 2000).

Todas as bactérias são organismos procariotos. Em sua estrutura, além da sua membrana plasmática, existe uma espessa e rígida camada chamada parede bacteriana envolvendo a membrana, e em alguns casos esta parede é envolvida por uma camada viscosa chamada cápsula. Devido à estrutura de suas paredes, a bactéria pode ser categorizada como Gram-Negativa ou Gram-Positiva, dependendo do seu comportamento diante da técnica de Gram. Quando, após a aplicação, as bactérias aparecerem coradas em roxo o resultado é positivo, caso contrário é negativo (JUNQUEIRA E CARNEIRO, 2000).

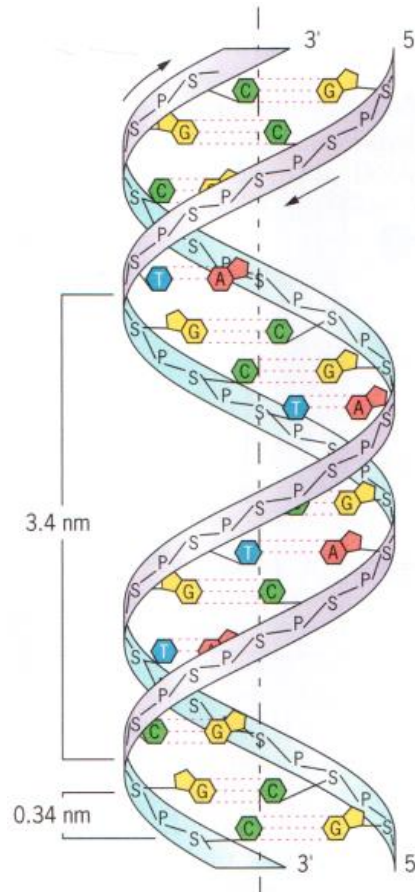
## 2.3 DNA E RNA

O DNA (Figura 2) é uma estrutura helicoidal em dupla hélice, composta por duas cadeias intercaladas de material estrutural chamados nucleotídeos. Cada nucleotídeo é composto por uma pentose (no DNA chamada de Desoxirribose), um grupo fosfato e uma base nitrogenada. Esta última pode ser dos tipos adenina (A), citosina (C), guanina (G) e timina (T) (GRIFFITHS *et al.*, 2002; ALBERTS *et al.*, 1999).

Os nucleotídeos em uma cadeia de DNA estão unidos através de pontes fosfodiéster entre o grupo fosfato e o grupo OH (hidroxila) do carbono 3' do nucleotídeo adjacente. As cadeias do DNA são mantidas juntas, de forma antiparalela, através de pontes de hidrogênio entre às bases nitrogenadas. Devido à estrutura química das bases, as ligações acontecem somente entre timina e adenina, e entre citosina e guanina (ZAHA, 1996).

---

<sup>2</sup> DNA: *deoxyribonucleic acid* (Ácido Desoxirribonucleico)



**Figura 2: Estrutura do DNA**  
**Fonte: (SNUSTAD E SIMMONS, 2003)**

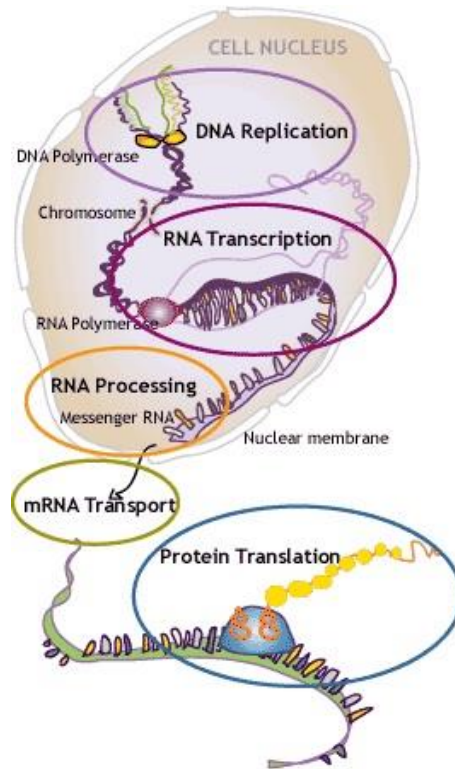
A molécula de RNA<sup>3</sup> é muito semelhante quimicamente à molécula de DNA. As principais diferenças são que o RNA é uma fita simples, a pentose no RNA é uma ribose e a base nitrogenada timina (T) do DNA é substituída pela uracila (U), que também pode parear com a adenina (A) (ALBERTS *et al.*, 1999).

Uma molécula de DNA contém informações para codificar milhares de proteínas. Cada seção do DNA é transcrita em um tipo de molécula de RNA, que posteriormente poderá ser traduzido em uma proteína. Cada uma dessas seções é denominada Gene. A sua descoberta, compreensão e função tem sido fonte de profunda percepção de dois dos maiores mistérios da biologia: o que faz uma espécie ser o que ela é, e o que causa a variação dentro de uma espécie (ALBERTS *et al.*, 2010; GRIFFITHS *et al.*, 2002).

<sup>3</sup> RNA: *ribonucleic acid* (Ácido Ribonucleico)

## 2.4 DOGMA CENTRAL DA BIOLOGIA MOLECULAR

O dogma central da biologia molecular (Figura 3) define os processos de replicação do DNA, de transcrição (fluxo da informação genética de DNA a RNA) e de tradução (fluxo da informação genética do RNA à proteína).



**Figura 3: Dogma Central da Biologia Molecular**

Fonte: NobelPrize.org<sup>4</sup>

### 2.4.1 Replicação

Cada uma das cadeias de nucleotídeos de uma molécula de DNA pode servir como molde para uma nova molécula de DNA através do processo de replicação (Figura 4). Neste processo, uma enzima chamada Helicase quebra as pontes de hidrogênio que unem as duas cadeias, iniciando em pontos específicos chamados de origens de replicação. Estas origens são marcadas por uma sequência específica de nucleotídeos e abrangem aproximadamente 100 pares de bases (ZAHA, 1996; ALBERTS, et al.,1999).

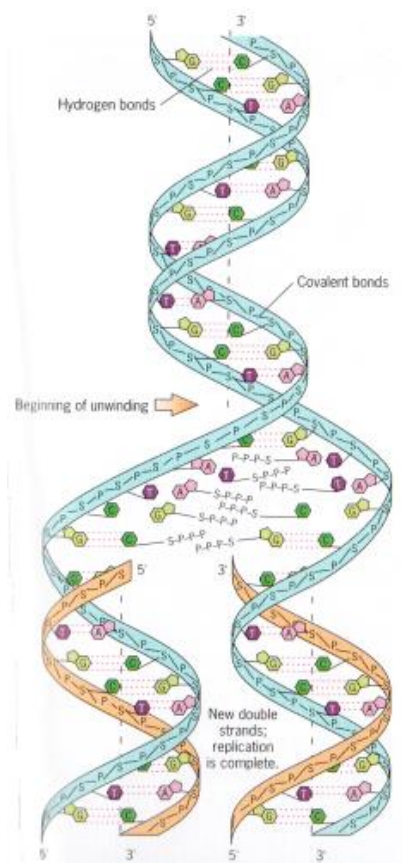
Os locais onde a molécula de DNA é aberta são chamados de Forquilhas de Replicação, e são nestes locais que o motor de replicação atua. O responsável pela síntese das novas moléculas de DNA é uma enzima chamada DNA polimerase. Ela percorre o DNA

<sup>4</sup> <http://www.nobelprize.org/educational/medicine/dna/index.html>



aberto, adicionando nucleotídeos a uma extremidade 3' de uma cadeia de nucleotídeos em crescimento através da formação de ligações fosfodiéster (ZAHA, 1996; ALBERTS *et al.*,1999).

A DNA polimerase pode dar continuidade a uma cadeia de DNA já existente, porém não pode iniciar uma nova. Para iniciar a síntese do novo DNA, é necessário um oligonucleotídeo curto (geralmente de um a sessenta nucleotídeos de comprimento) chamado *primer*, que se une a cadeia de DNA utilizada como molde formando uma cadeia dupla, permitindo que a DNA polimerase de continuidade ao processo de polimerização. Na bactéria *E.coli* são utilizado *primers* de RNA sintetizados por uma enzima chamada primase (GRIFFITHS *et al.*, 2002; JUNQUEIRA E CARNEIRO, 2000).

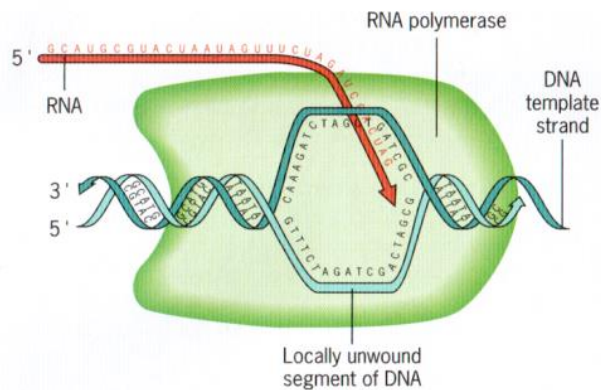


**Figura 4: Processo de replicação de DNA**  
**Fonte: (SNUSTAD E SIMMONS, 2003)**

## 2.4.2 Transcrição

No processo de transcrição (Figura 5), são sintetizados todos os RNAs necessários para os processos celulares e para a síntese de proteínas. Semelhante ao processo de

replicação, porém utilizando uma enzima chamada RNA polimerase. Esta enzima sintetiza quase todos os tipos de RNA em organismos procariontes. Na bactéria *E.coli*, a mesma é composta de quatro subunidades de pesos moleculares diferentes. A subunidade alfa ( $\alpha$ ) com peso molecular de 40.000 daltos, a subunidade beta ( $\beta$ ) com 150.000 daltos, a subunidade beta primo ( $\beta'$ ) com 160.000 daltos e a subunidade sigma ( $\sigma$ ) com 70.000 daltos (GRIFFITHS *et al.*, 2002; JUNQUEIRA E CARNEIRO, 2000).



**Figura 5: Processo de transcriao de DNA**  
**Fonte: (SNUSTAD; SIMMONS, 2003)**

Uma das cadeias do DNA  utilizada como molde e transcrita em uma cadeia de RNA, substituindo somente as bases timina (DNA) pela Uracila (RNA) (VALIATI, 2006; JUNQUEIRA E CARNEIRO, 2000).

O RNA mensageiro representa a transcriao de um segmento de uma das cadeias de uma molecula de DNA e  utilizado para transferir as informaoes geneticas do DNA para a sntese de proteinas. Ele  muito maior do que a proteina por ele gerada, por serem necessarios tres nucleotdeos para ser gerado um aminocido. Este grupo de tres nucleotdeos no mRNA  chamada de codon (ZAHA, 1996; JUNQUEIRA E CARNEIRO, 2000).

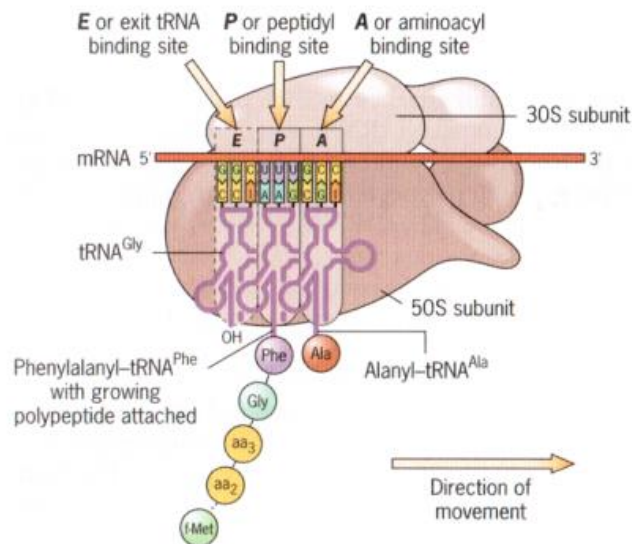
A funao do RNA transportador  transferir os aminocidos para as posioes corretas nas cadeias de aminocidos em formaao nos complexos de ribossomos e RNA mensageiro. Para isso, o tRNA  capaz de se combinar com aminocidos, e  capaz tambem de parear com os codons da molecula de mRNA atraves de tres grupos de bases chamadas anticodons (JUNQUEIRA E CARNEIRO, 2000).

O RNA ribossmico  muito mais abundante do que os RNAs citados anteriormente (cerca de 80% de todo RNA celular). Quando combinado com proteinas ribossmicas, formam particulas chamadas ribossomos, que tem como funao principal ligar-se a filamentos de mRNA para efetuar a sntese de proteinas (JUNQUEIRA E CARNEIRO, 2000).

### 2.4.3 Tradução

A tradução (Figura 6) é o processo de transformação das informações do mRNA em proteínas. Nos procariotos, para que esta síntese de proteínas tenha início, é necessário que ocorra uma ligação entre o mRNA e o Ribossomo. O ponto de ligação do Ribossomo ao mRNA é chamado de RBS<sup>5</sup>, que é identificado por duas características básicas (ZAHA, 1996):

- Existência do códon de iniciação, que geralmente é AUG, e raramente pode se manifestar como GUG ou UUG.
- Existência de uma sequência de nucleotídeos complementar chamada sequência de Shine-Dalgarno.



**Figura 6: Processo de tradução do mRNA**  
**Fonte: (SNUSTAD E SIMMONS, 2003)**

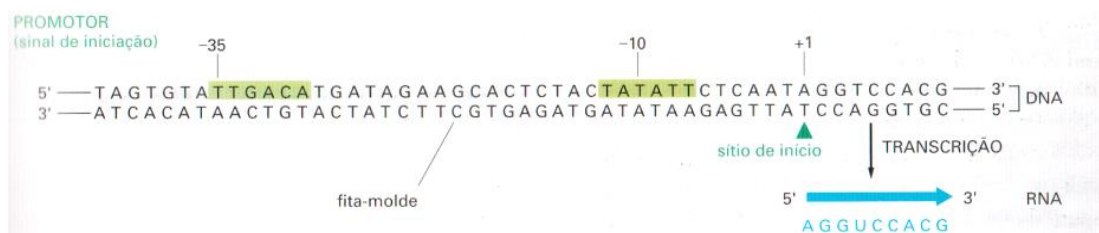
Durante a síntese de proteínas, o ribossomo desloca-se sobre o mRNA na direção 5' -> 3', capturando moléculas de tRNA complementares e gerando uma cadeia de aminoácidos. Cada códon na molécula de mRNA irá parear com um anticódon em uma molécula de tRNA, que em sua extremidade 3' terá acoplado o aminoácido. A ligação entre o tRNA e o aminoácido dependem da atuação de enzimas chamadas aminoacil-tRNA sintetase, onde cada aminoácido possui uma enzima correspondente (ALBERTS *et al.*, 1999).

<sup>5</sup> *Ribosomal Binding Site*: Sítio de ligação dos Ribossomos

## 2.5 PROMOTORES

Os promotores são sequências na cadeia de DNA, que indicam à RNA polimerase o ponto de início da síntese de RNA no processo de transcrição. O primeiro nucleotídeo a ser copiado é chamado de sítio de início, e é demarcado como +1 na molécula de DNA. Os nucleotídeos localizados na direção da extremidade 5', também conhecida como *upstream*, recebem uma numeração negativa e decrescente. Os nucleotídeos localizados na direção da extremidade 3', também conhecida como *downstream*, recebem um número positivo e crescente (ALBERTS *et al.*, 1999; ZAHA, 1996).

Os promotores estão localizados na porção anterior ao sítio de início, ou seja, na região intergênica (direção *upstream*), que possui a sequência do DNA que não será codificado. Na bactéria *E.coli*, os mesmos são caracterizados pelas sequências de seis nucleotídeos TTGACA e TATAAT (Figura 7) nas regiões -35 e -10 respectivamente. Estas sequências refletem os nucleotídeos mais comuns encontrados nestas posições na coleção de promotores (ALBERTS *et al.*, 2010).



**Figura 7: Promotor**  
**Fonte: (ALBERTS *et al.*, 1999)**

Na RNA polimerase a subunidade  $\sigma$ , também conhecida como fator sigma, pode desacoplar do restante da enzima. Quando completa, a mesma é conhecida como holoenzima de RNA polimerase. É a subunidade  $\sigma$  que permite que a enzima reconheça as regiões promotoras, e a mesma deve estar acoplada no início do processo de transcrição do DNA (GRIFFITHS *et al.*, 2002).

Na bactéria *E.coli*, quando a RNA polimerase entra em contato com a cadeia de DNA, a mesma efetua um acoplamento suave, formando uma estrutura chamada “complexo promotor fechado”. Quando localiza um promotor, a enzima aumenta a força do acoplamento, desfazendo a estrutura helicoidal e desnaturando o dúplex de DNA, caracterizando uma estrutura chamada “complexo promotor aberto”. Após iniciada a transcrição, a subunidade  $\sigma$  desacopla da RNA polimerase (GRIFFITHS *et al.*, 2002).

## 2.6 CONSIDERAÇÕES FINAIS

Todos os seres vivos são formados por estruturas menores denominadas células. A Biologia Molecular é a ciência que estuda a célula e os seus processos internos, em especial o material genético, suas características e funções. Através de seu estudo é possível entender os mecanismos que tornam os seres vivos diferentes entre si, mesmo dentro de uma mesma espécie (ZAHA, 1996).

A quantidade de dados gerados pela Biologia Molecular e o processamento necessário para analisar tais dados são muito grandes. Por este motivo surgiu a Bioinformática, ciência que une conhecimentos de Biologia Molecular e Informática, e os Bancos de Dados de Biologia Molecular (BDBM). No próximo capítulo, ambos os conceitos serão detalhados e alguns portais sobre Biologia Molecular existentes na *world wide web* serão apresentados.

### 3 BIOINFORMÁTICA

Com a descoberta da estrutura do DNA por Watson e Crick em 1953, e posteriormente do código genético e do fluxo de informações biológicas, tornou-se possível pesquisar as sequências monoméricas constituintes do DNA. Na década de 90, com o surgimento dos sequenciadores automáticos de DNA, recursos computacionais eficientes tornaram-se necessários, devido ao aumento exponencial da quantidade de sequências a serem armazenadas e processadas. Em abril de 2007, os bancos de dados de sequências de nucleotídeos ultrapassavam os 100 bilhões de bases, e o banco de dados de estruturas de macromoléculas ultrapassava as 39.000 entradas apenas para proteínas, ou seja, um grande volume de dados a ser processado. Devido a esta necessidade de melhorar cada vez mais os recursos computacionais para armazenamento e processamento de dados relacionados à Biologia Molecular surgiu a Bioinformática, que é uma ciência que reúne o conhecimento de ambas as áreas (LESK, 2008; PROSDOCIMI *et al.*, 2003).

A seguir, serão apresentadas algumas ferramentas de Bioinformática aplicadas ao uso da Biologia Molecular.

#### 3.1 BANCOS DE DADOS DE BIOLOGIA MOLECULAR

Segundo Lesk (2008)

Um banco de dados inclui arquivos de informações, uma organização lógica dessas informações e ferramentas para se ter acesso a elas. Os Bancos de Dados de Biologia Molecular contêm sequências de ácidos nucleicos e de proteínas, estruturas e funções de macromoléculas, padrões de expressão, redes de vias metabólicas e cascatas de regulação.

Os BDBMs que armazenam informações sobre sequências de nucleotídeos, aminoácidos ou estruturas de proteínas podem ser classificados em primários e secundários. Os BDBMs primários armazenam dados ainda não processados. Os seus principais serviços são o *GenBank*, o EBI (*European Bioinformatics Institute*), o DDBJ (*DNA Data Bank of Japan*) e o PDB (*Protein Data Bank*). Os secundários, como o PIR (*Protein Information Resource*) ou o SWISS-PROT, são bancos de dados criados sobre as informações dos primários, envolvendo processamento e análise (PROSDOCIMI *et al.*, 2003).

Para que estas informações sejam úteis, é necessário que o banco de dados tenha uma estrutura de dados organizada e concisa, caso contrário o mesmo acaba se tornando um cemitério de informações. Para garantir esta integridade, os administradores do BD formatam

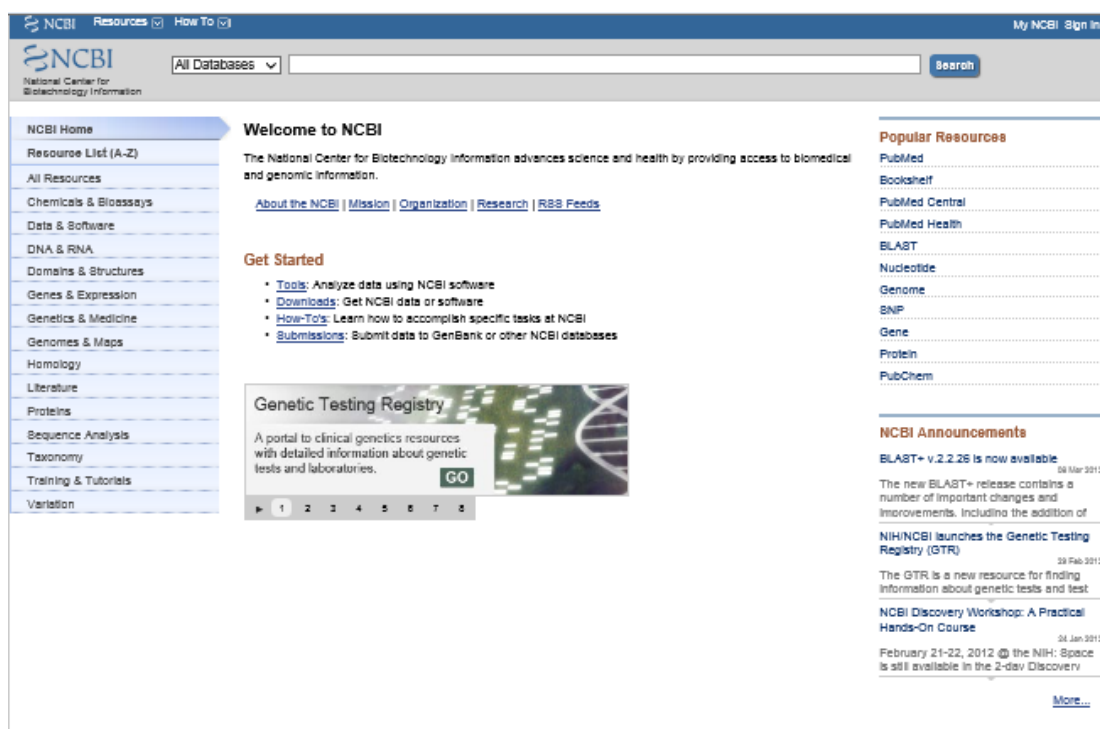
os dados experimentais postados pelos cientistas de forma adequada, e em seguida adicionam os mesmos ao domínio público do Banco de Dados (LESK, 2008).

## 3.2 PORTAIS

Nas próximas seções serão apresentados alguns portais<sup>6</sup> *web* que fornecem ferramentas e métodos para acesso as informações contidas em seus BDBMs.

### 3.2.1 NCBI

O NCBI<sup>7</sup> (Figura 8) surgiu a partir de uma iniciativa do senador Claude Pepper, que entendeu a importância dos métodos de processamento de dados computadorizados para a realização de pesquisas biomédicas e patrocinou a legislação que deu origem ao centro, no dia 4 de novembro de 1988, como uma divisão da Biblioteca Nacional de Medicina (NLM<sup>8</sup>) no Instituto Nacional de Saúde (NIH<sup>9</sup>).



**Figura 8: Portal NCBI**  
**Fonte: National Center for Biotechnology Information<sup>10</sup>**

<sup>6</sup> Portal: *Site de Internet* que disponibiliza uma série de serviços ou *subsites* que manipulam e proveem informações sobre determinado assunto: Como motores busca, notícias e *forums*.

<sup>7</sup> NCBI: *National Center for Biotechnology Information*

<sup>8</sup> NLM: *National Library of Medicine*

<sup>9</sup> NIH: *National Institute of Health*

<sup>10</sup> <http://www.ncbi.nlm.nih.gov/>

O centro tem por missão desenvolver novas tecnologias para auxiliar na compreensão de processos moleculares e genéticos que controlam a saúde e as doenças. O NCBI é responsável pela criação e manutenção de sistemas automatizados para armazenamento e processamento de conhecimentos sobre biologia, bioquímica e genética.

O NCBI fornece uma série de bancos de dados para a comunidade médica e científica. Dentre eles pode ser citado o *Genbank*, que é um banco de dados de sequências genéticas e coleção anotada de todas as sequências disponíveis publicamente sob responsabilidade do NIH. Possui aproximadamente 126 bilhões de bases em 135 milhões de sequências nas divisões tradicionais do *GenBank* e 191 bilhões de bases em 62 milhões de sequências na divisão WGS<sup>11</sup> (*Whole Genome Shotgun*). O mesmo foi projetado para possibilitar o acesso a toda a comunidade científica com as informações mais atualizadas sobre sequências de DNA.

É possível efetuar buscas nas bases de dados do *GenBank* de várias formas diferentes: Seja através do *EntrezNucleotide*<sup>12</sup> para buscar identificadores de sequências e anotações; do BLAST<sup>13</sup> para pesquisar e alinhar sequências do *GenBank* em uma consulta; ou pesquisar, vincular, e efetuar o *download* de sequências programaticamente utilizando as ferramentas do NCBI *e-utilities*<sup>14</sup>.

O *GenBank* permite efetuar a atualização de uma série de informações como sequências de mRNA, genes procarióticos, genes eucarióticos, rRNA, sequências virais, Transposons ou sequências de inserção, sequências microssatélite, pseudogenes, vetores de clonagem, e filogenética ou conjuntos populacionais.

Para acessar de forma eficiente as informações contidas no *GenBank* e nos demais bancos de dados mantidos pelo NCBI é utilizado o Entrez<sup>15</sup>, que é o sistema de busca primário do NCBI e quase todas as caixas de pesquisa encontradas no site do NCBI acessam o mesmo. Ele integra o banco de dados de literatura biomédica do Pubmed com outros 39 bancos de dados de literatura e de informações biomoleculares. Novos bancos de dados são adicionados com o passar do tempo, dependendo dos avanços das ciências biomédicas e dos novos tipos de dados que surgem destas pesquisas.

Diversas outras ferramentas estão disponíveis no portal do NCBI, como o BLAST<sup>16</sup> (*Basic Local Alignment Search Tool*) que é uma ferramenta utilizada para localizar regiões de

---

<sup>11</sup> <http://www.ncbi.nlm.nih.gov/genbank/wgs>

<sup>12</sup> <http://www.ncbi.nlm.nih.gov/nucleotide/>

<sup>13</sup> <http://blast.ncbi.nlm.nih.gov/Blast.cgi>

<sup>14</sup> <http://www.ncbi.nlm.nih.gov/books/NBK25500/>

<sup>15</sup> <http://www.ncbi.nlm.nih.gov/gquery/>

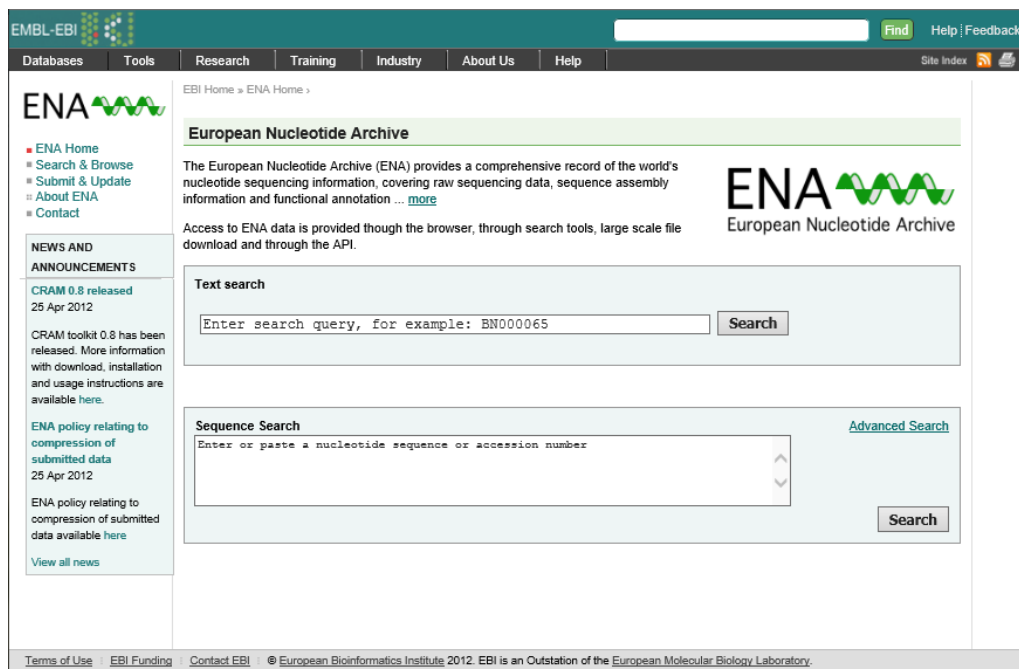
<sup>16</sup> [http://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE\\_TYPE=BlastHome](http://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastHome)



similaridade entre sequências de nucleotídeos ou proteínas (o programa compara sequências de nucleotídeos ou proteínas e calcula a significância estatística dos resultados). E o PubMed<sup>17</sup> que é um portal *web* desenvolvido e mantido pelo NCBI de forma gratuita, que possui mais de 21 milhões de citações sobre biomedicina do *MEDLINE*<sup>18</sup>, periódicos sobre ciências da vida e livros *on-line*. Provê também acesso a *web sites* e *links* de outras fontes sobre Biologia Molecular do NCBI.

### 3.2.2 ENA

O *European Nucleotide Archive* (ENA) (Figura 9) é um portal que provê informações sobre sequências de nucleotídeos, desenvolvido e mantido pelo EMBL-EBI<sup>19</sup> (*European Bioinformatics Institute*) sob a orientação do Comitê Consultivo Internacional do INSDC<sup>20</sup> (*International Nucleotide Sequence Database Collaboration*) e um recém-formado Conselho Consultivo Científico. Apesar de ter aproximadamente 30 anos de existência, os serviços do ENA mudam constantemente com o intuito de refletir o crescente volume de dados, sempre aprimorando a tecnologia de sequenciamento e ampliando as aplicações onde são armazenadas.



**Figura 9: Portal ENA**  
**Fonte: European Nucleotide Archive<sup>21</sup>**

<sup>17</sup> <http://www.ncbi.nlm.nih.gov/pubmed>

<sup>18</sup> <http://www.nlm.nih.gov/pubs/factsheets/medline.html>

<sup>19</sup> <http://www.ebi.ac.uk/>

<sup>20</sup> <http://www.insdc.org/>

<sup>21</sup> <http://www.ebi.ac.uk/ena/>

O ENA é formado por uma série de bancos de dados incluindo o *EMBL Nucleotide Sequence Database*, também conhecido como *EMBL-bank*, o recentemente estabelecido *Sequence Read Archive (SRA)*<sup>22</sup> e o *Trace Archive*, cada um com seus formatos e padrões.

Os dados do ENA podem ser acessados através do próprio portal, de ferramentas de pesquisa, de *download* de arquivos em larga escala e da API do próprio portal. A API do portal é chamada *ENA Browser* que é disponibilizada através do protocolo REST para *webservices*. O ENA ainda disponibiliza um serviço de FTP<sup>23</sup> para possibilitar o *download* dos arquivos.

É possível também submeter dados ao ENA. Quando submetidos, eles podem ser mantidos em sigilo até o trabalho associado ser publicado. O ENA atribui números de adesão a todos os dados submetidos, que devem ser citados em todos os trabalhos onde os mesmos aparecerem como parte da análise. Quando os dados se tornam públicos, os dados são trocados com o NCBI e com o DDBJ como parte do acordo de troca de dados do INSDC.

### 3.2.3 DDBJ

O *DNA Data Bank of Japan (DDBJ)* (Figura 10) é o único banco de dados de sequências de nucleotídeos da Ásia. É oficialmente certificado para coletar dados de pesquisadores e emitir o número de acesso internacionalmente reconhecido para provedores de dados. O principal objetivo do DDBJ é melhorar a qualidade do *International Nucleotide Sequence Database (INSD)*. Quando pesquisadores disponibilizam dados através do INSD, o DDBJ procura descrever as informações sobre os dados da forma mais rica possível, de acordo com as regras unificadas do INSD.

Os dados submetidos ao DDBJ são processados e disponibilizados utilizando o seu formato proprietário de distribuição (*flat file*). O arquivo texto inclui a sequência de nucleotídeos, as informações do provedor, referências, os organismos-fonte e as características (definidas pela tabela de definição de características do DDBJ/EMBL/*GenBank* para descrever a natureza biológica).

---

<sup>22</sup> <http://www.ncbi.nlm.nih.gov/sra/>

<sup>23</sup> <ftp://ftp.ebi.ac.uk/pub/databases/embl/>

**Figura 10: Portal DDBJ**  
**Fonte: DNA Data Bank of Japan<sup>24</sup>**

O portal não possui um método de pesquisa direta como os citados até o momento, porém possui uma série de ferramentas para busca e análise.

<sup>24</sup> <http://www.ddbj.nig.ac.jp/>

### 3.2.4 RegulonDB

O RegulonDB (Figura 11) é um portal que provê informações sobre o mecanismo de transcrição do DNA da bactéria *Escherichia Coli K12*. O portal foi fundado pelo NIH e pelo CONACyT<sup>25</sup> em 1998.



Figura 11: Portal RegulonDB

Fonte: RegulonDB<sup>26</sup>

O portal disponibiliza um mecanismo de consulta direta através de uma caixa de texto livre permitindo filtros por: Gene, *Gensor*<sup>27</sup> *Unit*, Operon, Regulon, Sigmulon, sRNA<sup>28</sup> ou condições de crescimento. O mesmo permite o *download* de uma série de informações como promotores, fatores de transcrição e operons, e também possui uma série de outras ferramentas com finalidades diversas como:

- *RegulonDB Overviews*: Possui uma coleção de informações disponibilizada em 5 categorias maiores: Fator de Transcrição, Operon, Fator Sigma, Regulon e Sítio de Ligação do DNA.

<sup>25</sup> Consejo Nacional de Ciencia y Tecnología (México)

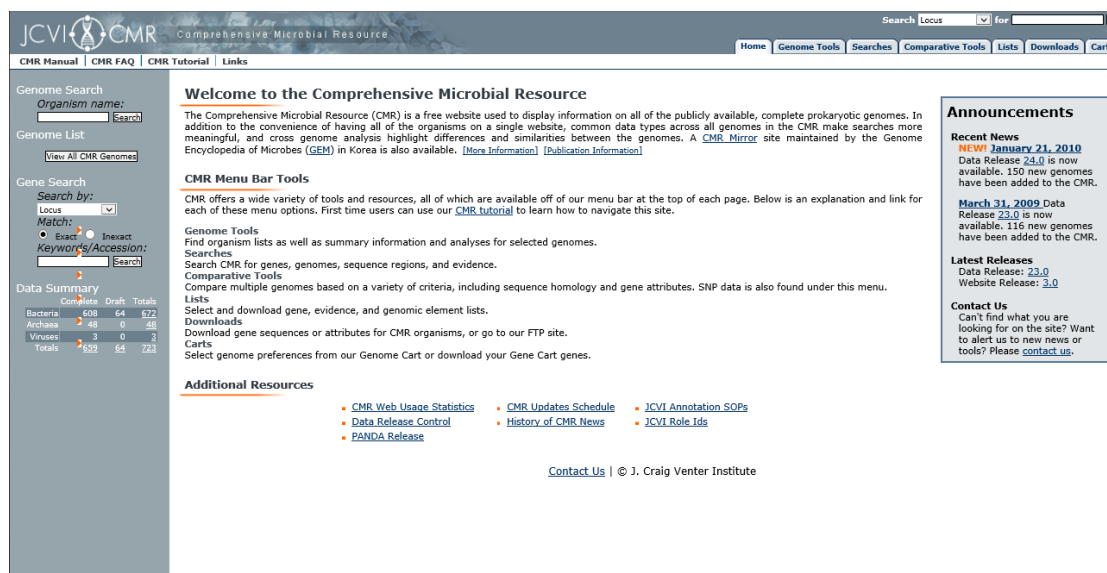
<sup>26</sup> <http://regulondb.ccg.unam.mx/>

<sup>27</sup> *Genetic Sensory Response*: Resposta Sensorial Genética

<sup>28</sup> Small RNA

- *RSA Tools*: Disponibiliza uma série de ferramentas específicas para a detecção de sinais regulatórios em sequências não codificadas.
- *GET Tools*: Ferramenta para análise da regulação da transcrição dos experimentos efetuados sobre expressão gênica da bactéria *E.coli* em âmbito global.
- *Simple Retrieval*: Ferramenta que possibilita pesquisas mais avançadas, permitindo filtros por até quatro categorias diferentes, e em que dados a pesquisas deve ser aplicada.
- *Promoter Analysis Tool*: Ferramenta para análise de regiões promotoras.
- *Web Services*: Permite o acesso as informações do *RegulonDB* utilizando o protocolo SOAP.
- *Regulatory Networks*: Permite extrair uma série de circuitos de interação, gerando informações sobre redes regulatórias.
- *DrawingTracesTool*: Ferramenta que permite a geração de imagens relacionadas ao DNA, envolvidas na regulação gênica.

### 3.2.5 JCVI-CMR



**Figura 12: Portal JCVI-CMR**  
**Fonte: JCVI-CMR Comprehensive Microbial Resource<sup>29</sup>**

O *Comprehensive Microbial Resource* (CMR) (Figura 12) é um portal mantido pelo *J. Craig Venter Institute*<sup>30</sup> (JCVI), que tem por objetivo permitir aos usuários acessarem todas as

<sup>29</sup> <http://cmr.jcvi.org/tigr-scripts/CMR/CMrHomePage.cgi>

<sup>30</sup> <http://www.jcvi.org/>

informações sobre sequências de genomas bacterianos finalizados até a data da consulta, bem como qualquer subconjunto dos mesmos. Todas estas informações são armazenadas em um banco de dados relacional chamado Omniome, que além das sequências, possui uma série de outras informações como anotações para cada genoma, dados de identificação de outros portais (*GenBank* e *SwissProt*) e dados de homologia pré-computadas.

O CMR disponibiliza uma série de ferramentas para acesso as informações de sua base. Além de uma série de ferramentas específicas para análise e consulta, no topo do portal existe uma ferramenta de pesquisa que permite efetuar consultas em todas as anotações do CMR, genomas, propriedades dos genomas, símbolos dos genes, ferramentas, nome e também texto livre.

### 3.3 CONSIDERAÇÕES FINAIS

Atualmente existem diversos portais que disponibilizam dados, processados ou não, sobre Biologia Molecular. Os portais apresentados neste capítulo possuem diversos métodos de busca e recuperação de resultados, alguns deles também permitem o envio de dados para serem incorporados aos seus BDBMs. Na Tabela 1 é apresentado um resumo das principais funcionalidades destes portais.

**Tabela 1: Requisitos localizados nos portais**

Requisitos		NCBI (Entrez)	ENA	DDBJ(ARSA)	RegulonDB	JCVI-CMR
Busca Organismo	Nome	X	X	X		X
	Reino	X	X	X		X
	Família	X	X	X		X
Busca Gene	Nome	X	X	X	X	X
	Símbolo	X	X	X	X	X
	Função	X	X	X		X
	%GC					
Busca Região Intergênica	Tamanho					
	Faixa de Tamanho					
	Posição na fita					
	DNA fita 5' ou 3'		X			
	Fatias da Região		X			
	Gene				X	
Busca Método de Predição	Molécula de DNA					X
	Nome	X				
Download	% Acerto					
	Txt	X	X		X	X
	Html	X	X		X	
Upload	Xml	X	X			
			X	X		

Mesmo com esta diversidade de portais disponíveis, nenhum contempla as necessidades apresentadas pelo grupo de Bioinformática da Universidade de Caxias do Sul no estudo de promotores. Diante disto, um novo portal foi projetado e desenvolvido por dois alunos do Curso de Ciência da Computação na UCS em seus trabalhos de conclusão de curso. O histórico de desenvolvimento deste portal é apresentado no próximo capítulo.

## 4 PORTAL INTERGENICDB

Mesmo com uma série de portais e ferramentas disponíveis na *world wide web*, as necessidades do grupo de Bioinformática da UCS demandaram a criação de uma nova ferramenta para pesquisa e armazenamento de informações sobre Biologia Molecular.

O desenvolvimento desta nova ferramenta teve início no ano de 2009, pelo aluno Aurione Francisco Molin (MOLIN, 2009), com a proposta para o protótipo de um novo modelo de banco de dados chamado “PromotoresDB”. Em 2010, a aluna Vanessa Davanzo (DAVANZO, 2010), utilizou este modelo para desenvolver a base de dados para o portal “IntergenicDB”. Os mesmos serão apresentados nas próximas seções deste capítulo.

### 4.1 PROMOTORESDB

O banco de dados PromotoresDB foi projetado com o intuito de suprir a necessidade complementar do trabalho de doutorado, concluído em 2011, da professora e doutora Sheila de Ávila e Silva, intitulado “Redes neurais artificiais aplicadas no reconhecimento de regiões promotoras” (MOLIN, 2009; DAVANZO, 2010).

Seu objetivo principal era o de armazenar informações sobre sequências intergênicas com possível função biológica de DNA de organismos procariontes. A lista a seguir apresenta os principais requisitos e conceitos definidos para o desenvolvimento deste banco de dados (MOLIN, 2009; DAVANZO, 2010):

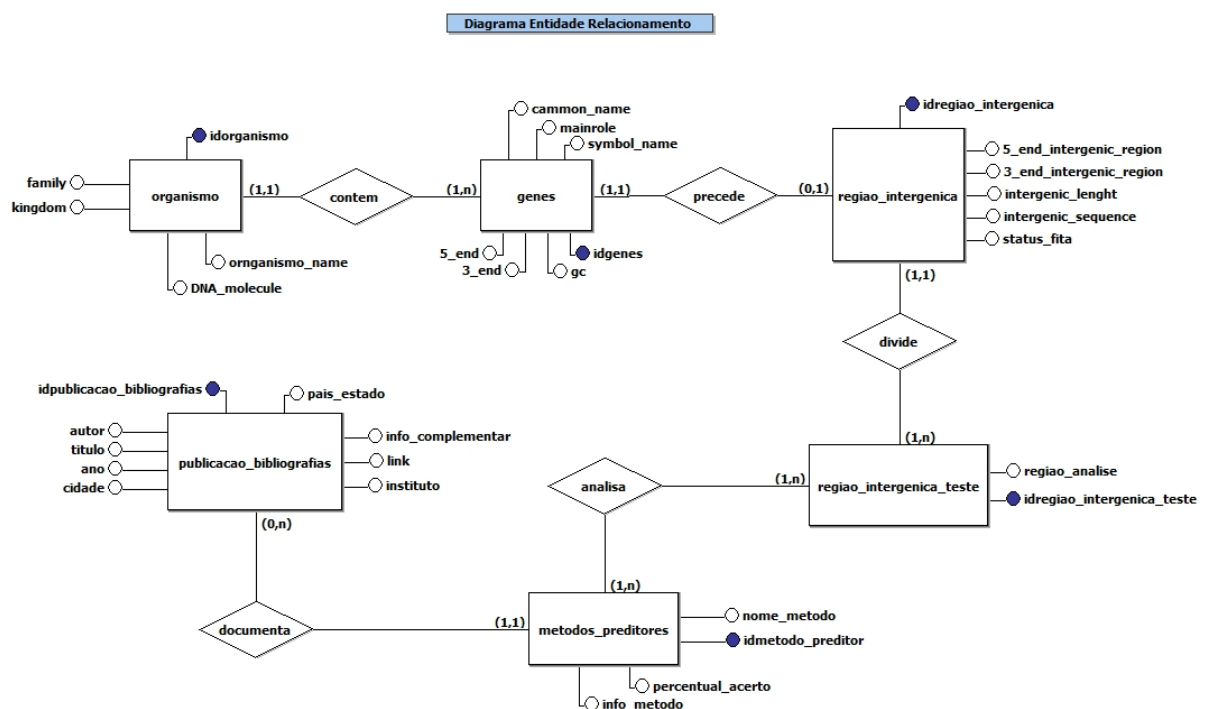
- Uma região promotora está vinculada a um organismo, e o mesmo possui informações como nome de identificação, reino, família e tipo de molécula ao qual ele pertence;
- Um gene possui um nome de identificação, um símbolo, o número da posição de início (5’), o número da posição de fim (3’), função, quantidade de nucleotídeos “GC” na sequência do gene;
- Uma região intergênica tem como informações: uma posição de início, uma posição de fim, o tamanho da sequência, a sequência de nucleotídeos e o tipo de fita;
- Uma região intergênica de teste possui 60 nucleotídeos, e é utilizada para a predição de sequências candidatas a serem promotoras ou não;
- Um método de predição é um algoritmo é utilizado na predição de sequências promotoras, e possui um nome de identificação, o percentual de uma cadeia



analisada ser promotora e uma descrição de suas características ou aplicabilidade;

- Uma publicação (artigos, trabalhos, etc.) está relacionada à área de estudo de promotores, e possui informações bibliográficas como: o autor, o título da obra, data de publicação, instituto vinculado, cidade e país, endereço *web*, e informações adicionais.

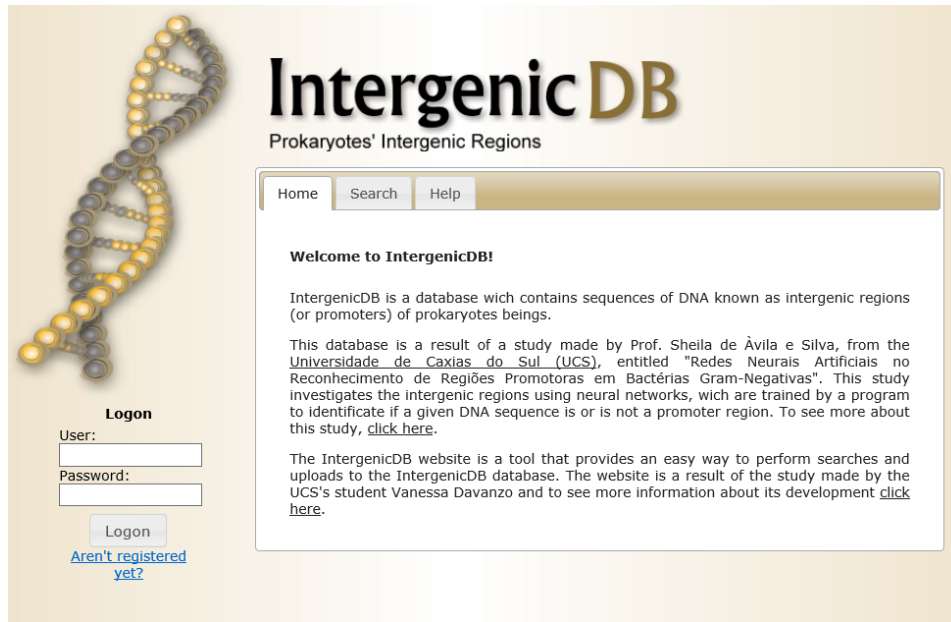
Diante destes requisitos foi possível elaborar o projeto do banco de dados. Abaixo, segue o modelo conceitual do protótipo desenvolvido por Molin:



**Figura 13: Modelo Conceitual do protótipo PromotoresDB**  
**Fonte: (MOLIN, 2009)**

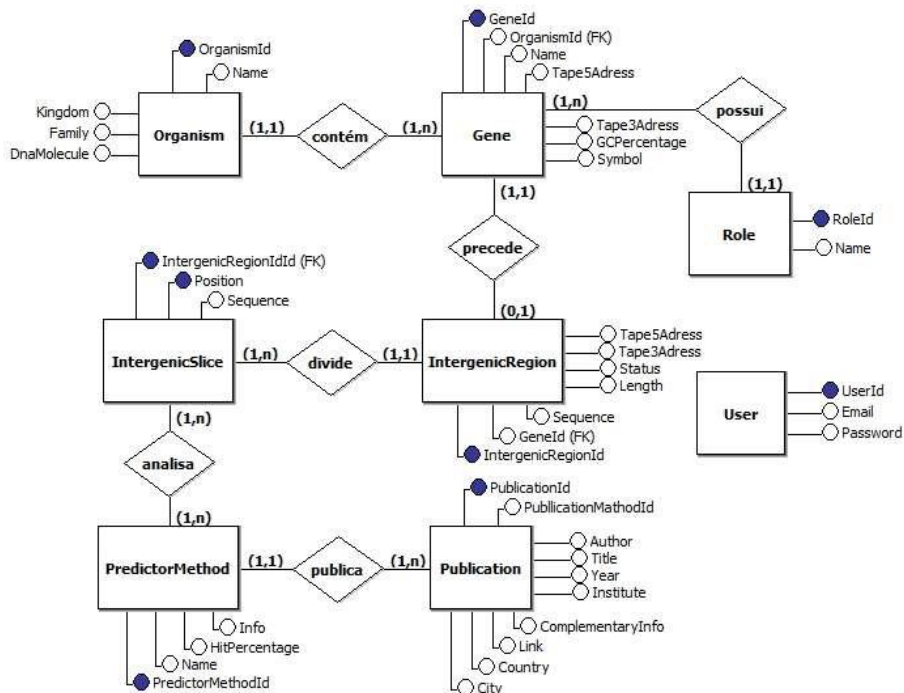
## 4.2 INTERGENICDB

O desenvolvimento do protótipo do PromotoresDB foi apenas parte do processo. Em um segundo momento, se fez necessária a criação de uma ferramenta para interagir com o mesmo, permitindo pesquisas ao banco de dados, *upload* e *download* de arquivos. Foi com este propósito que a aluna Vanessa Davanzo (DAVANZO, 2010) deu origem ao portal IntergenicDB (Figura 14).



**Figura 14: Portal IntergenicDB**  
**Fonte: (DAVANZO, 2010)**

O modelo do banco de dados promotoresDB sofreu algumas alterações para ser utilizado pelo IntergenicDB, foram incluídos algumas tabelas novas como a *User* e a *Role*, alguns nomes de entidades foram traduzidos para o inglês, alguns atributos tiveram seus nomes modificados e alguns relacionamentos foram criados. O modelo conceitual atualizado pode ser visto na figura 15.



**Figura 15: Modelo Conceitual do Banco de Dados**  
**Fonte: (DAVANZO, 2010)**

Os requisitos do IntergenicDB foram obtidos através de entrevistas com o usuário. As operações definidas foram (DAVANZO, 2010):

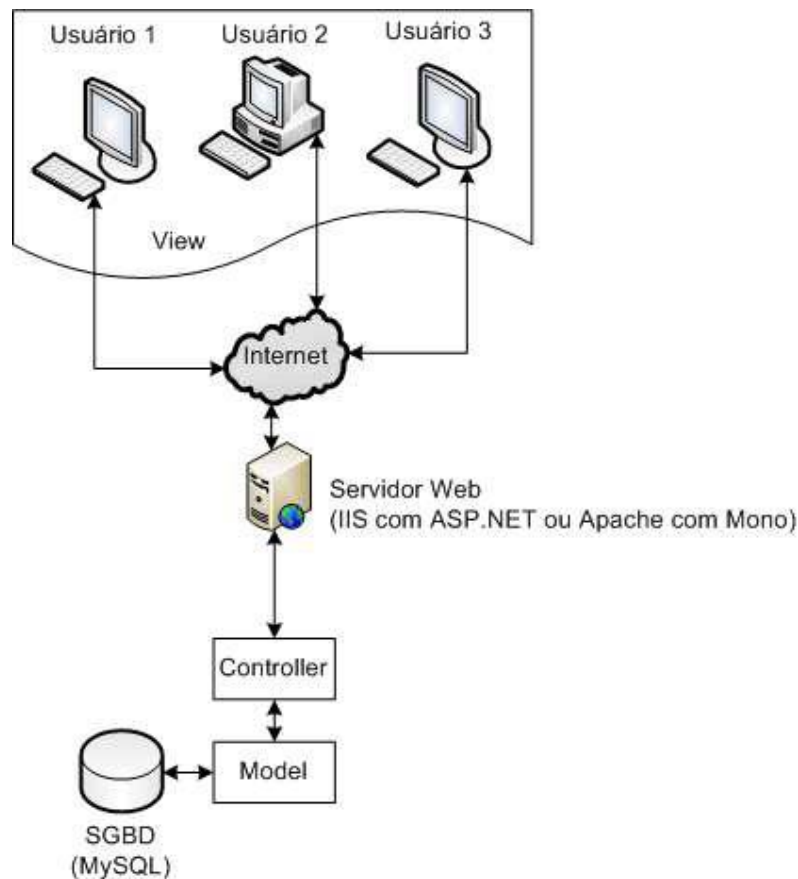
- *Busca por região intergênica*: Permite buscar por região intergênica, onde é possível pesquisar por tamanho, faixa de tamanho, *status* da fita e posição na fita;
- *Busca por gene*: Permite buscar por gene, onde é possível pesquisar por nome, símbolo, *mainrole* e porcentagem de GC;
- *Busca por fatia de região intergênica*: Permite efetuar a busca no sistema por gene, onde é possível informar o tamanho, posição na fita, buscar a fita *forward* ou *reverse*, e por fatias;
- *Busca por organismo*: Permite buscar por organismo, onde é possível pesquisar por nome, reino e família;
- *Download de arquivos*: Permite que o usuário efetue *download* dos resultados em formato TXT, XML e HTML;
- *Upload de arquivos*: Permitir que usuários cadastrados no sistema possam efetuar upload de informações para serem armazenadas no banco de dados.

Segundo Davanzo (2010), “a arquitetura do sistema é simples, sendo necessária apenas uma máquina servidora *Web* com a linguagem ASP.NET e o SGBD *MySQL*. O servidor recebe as requisições de serviços, as encaminha para o *Controller*, que consulta o *Model* (modelo de dados) e retorna a página em que o usuário irá interagir com as informações armazenadas no banco de dados, a *View*” (Figura 16).

Tendo em mãos o modelo do banco de dados e também os requisitos principais, foi desenvolvido o modelo de classes conceituais do sistema. Neste processo foram utilizados alguns padrões de projeto para resolver alguns problemas conhecidos. Os principais padrões utilizados foram (DAVANZO, 2010):

- *MVC*: É um dos padrões de projeto arquiteturais mais conhecidos em todo o mundo. Basicamente divide a aplicação em três camadas: *Model* (representa informações sobre o domínio de negócio), *View* (representação visual do domínio na interface com o usuário) e *Controller* (Manipula e controla as interações entre a interface com o usuário e o modelo de dados). O principal benefício deste padrão é a redução do acoplamento entre a camada de dados (*Model*) e a camada de apresentação (*View*) (FOWLER *et al.*, 2002);

- *Abstract Factory*: Um padrão de projeto de criação que permite criar objetos relacionados ou dependentes sem a necessidade de informar suas classes concretas. Alguns dos benefícios deste padrão são: O isolamento das classes concretas e a facilidade de mudança na família de produtos criados pela *Abstract Factory* (GAMMA *et al.*,1994);
- *Table Data Gateway*: Um padrão de projetos de acesso a dados que encapsula as operações SQL para acessar uma determinada tabela ou visão. (FOWLER *et al.*, 2002);



**Figura 16: Arquitetura do sistema**  
**Fonte: (DAVANZO, 2010)**

Na figura 17 está representado o diagrama de classes completo, incluindo os padrões de projeto.

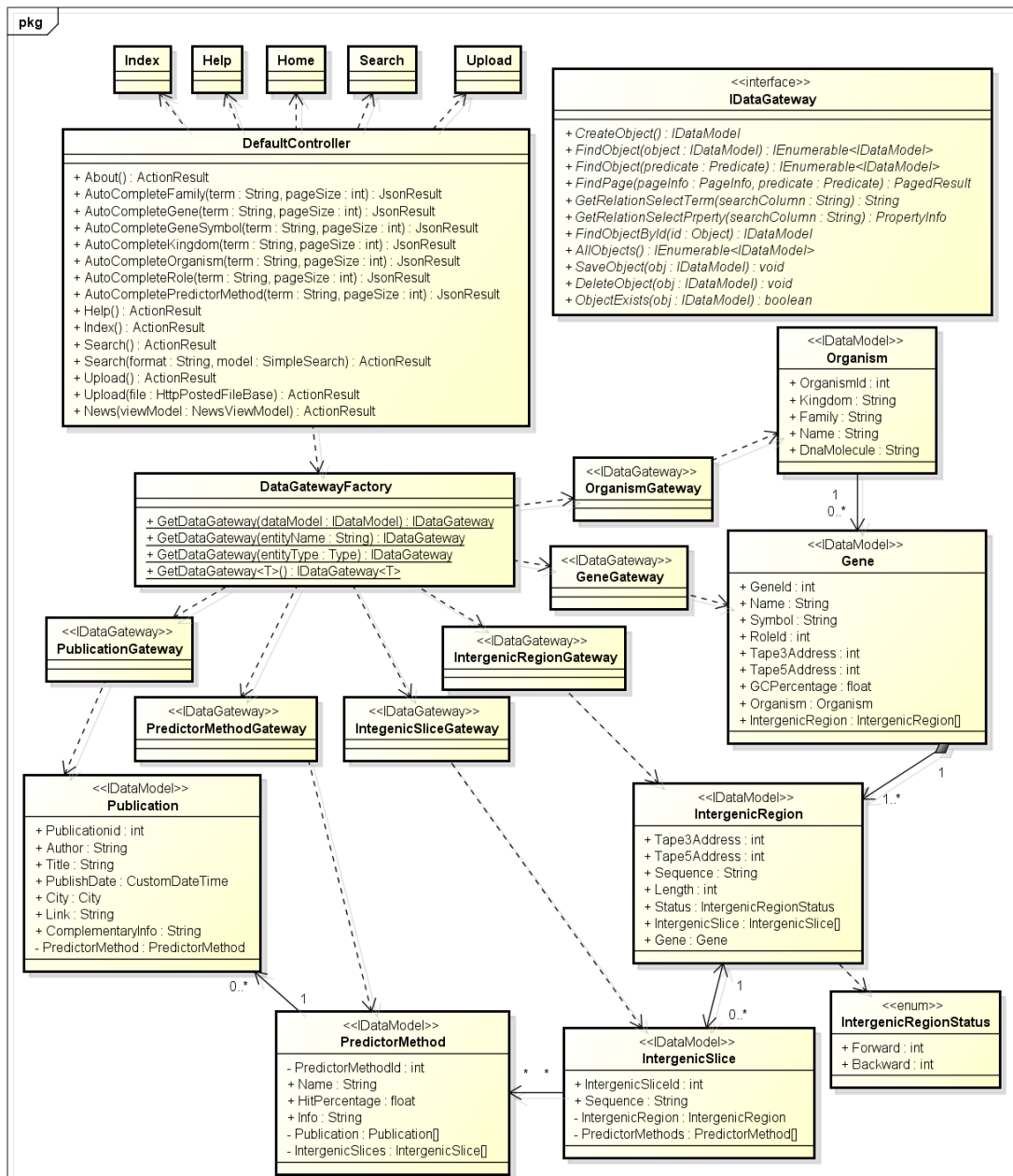
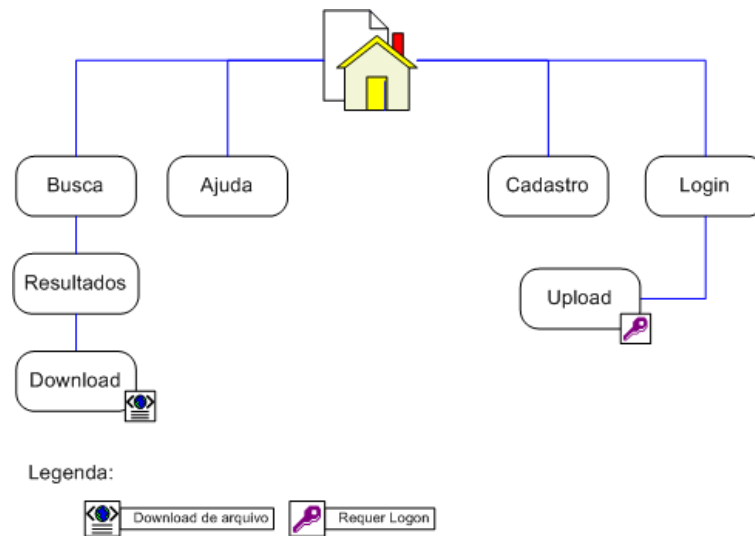


Figura 17: Diagrama de Classes

Para possibilitar a execução das consultas, a aluna Vanessa Davanzo (DAVANZO, 2010) desenvolveu um *framework* chamado *LP.Framework* que cria os predicados necessários de acordo com os parâmetros informados na pesquisa, traduz para um *script* SQL e executa o mesmo no banco de dados. O diagrama de classes deste framework pode ser visualizado no ANEXO A.

O sistema desenvolvido possui uma página inicial (*Home*) e outros quatro grupos de navegação (Figura 18). São eles:



**Figura 18: Modelo de Navegação**  
**Fonte: (DAVANZO, 2010)**

- *Home*: Página principal do portal. Permite acesso aos demais grupos de navegação e possui informações sobre os conceitos e objetivos do Portal;
- *Busca*: Possui todas as possibilidades de pesquisa definidas nos requisitos, possibilitando efetuar o *download* dos resultados em formato TXT e XML. Por ser a principal funcionalidade do portal, a sua interface está demonstrada na (Figura 19);

**Figura 19: Tela de busca do IntergenicDB**  
**Fonte: (DAVANZO, 2010)**

- *Ajuda*: Possui informações para auxiliar na utilização do sistema;
- *Cadastro*: Possibilita o cadastramento de usuários para que o mesmo efetue upload de arquivos;
- *Login*: Efetua o *login* no sistema, apenas usuários autenticados podem efetuar upload de arquivos.

Em relação às tecnologias, o banco de dados selecionado foi o *MYSQL* que além de ser uma ferramenta gratuita, é portátil e eficiente. No desenvolvimento do portal, a tecnologia utilizada foi o ASP.NET MVC<sup>31</sup> da *Microsoft* por sua produtividade e versatilidade, e a linguagem de programação utilizada foi o C# (MOLIN, 2009; DAVANZO, 2010).

#### 4.3 CONSIDERAÇÕES FINAIS

Para ser concebido o IntergenicDB passou por uma série de pesquisas efetuadas pelos alunos Aurione (MOLIN,2009) e Vanessa (DAVANZO,2010) em seus trabalhos de conclusão de curso na Universidade de Caxias do Sul. Destas pesquisas foram extraídos os requisitos mínimos para atender as necessidades específicas do grupo de bioinformática da universidade para a criação do portal. Contudo, para habilitar o IntergenicDB para ser utilizado em produção, são necessários que alguns novos requisitos sejam implementados. Estes requisitos são o foco principal deste trabalho e estão definidos no próximo capítulo.

---

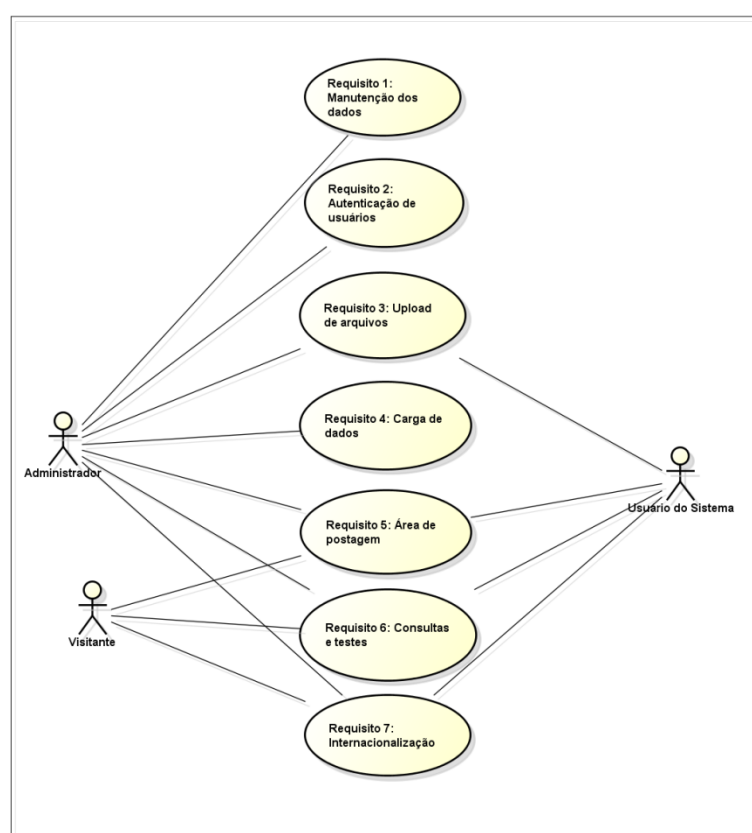
<sup>31</sup> <http://www.asp.net/mvc>

## 5 INTERGENICDB V2.0

Neste capítulo será efetuada a análise dos requisitos definidos pela profa. Scheila de Ávila e Silva que irão contemplar a nova versão do IntergenicDB. Serão mostradas também as ferramentas que serão utilizadas para o desenvolvimento destes requisitos.

### 5.1 REQUISITOS

Nesta seção serão apresentados os requisitos definidos em conjunto com a Profa. Dra. Scheila de Ávila e Silva, bem como a solução adotada para cada item. O caso de uso da figura 20 apresenta os requisitos e os atores envolvidos.



**Figura 20: Casos de uso das novas funcionalidades**

#### 5.1.1 Manutenção dos dados

Atualmente o portal IntergenicDB possui uma série de opções de busca de informações referentes a regiões promotoras. Contudo não existe uma forma simples de atualizar estas informações, as únicas opções disponíveis atualmente são o *upload* de arquivos e a atualização direta no SGBD. A utilização destas opções se torna inviável principalmente em casos onde a atualização deve ser efetuada sobre uma quantidade pequena de dados, ou



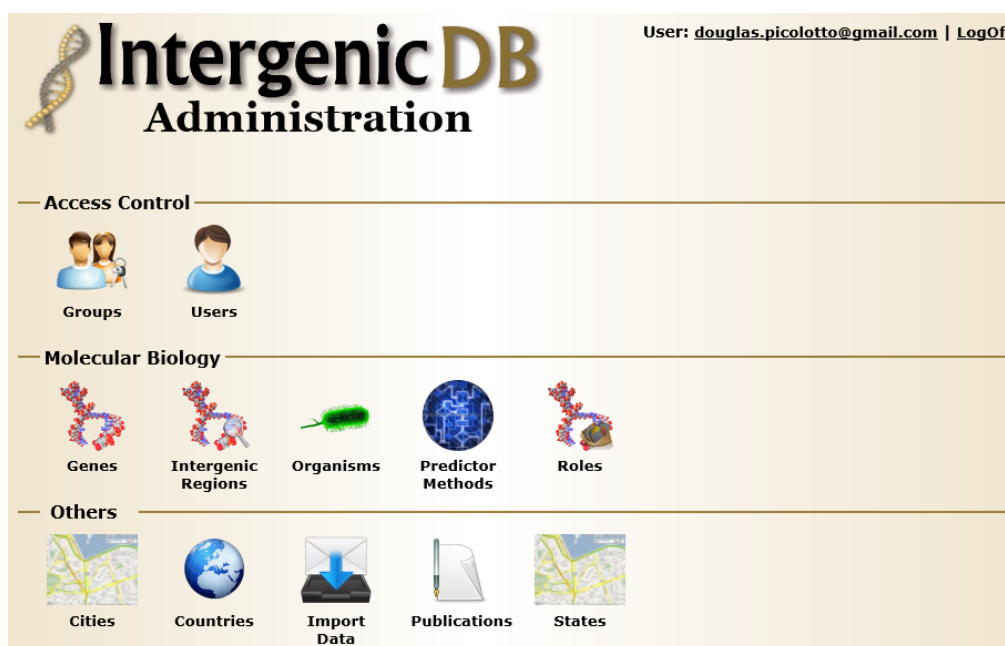
quando um usuário comum, sem conhecimentos avançados de informática, necessita efetuar alguma atualização.

Como solução, será criada uma área de administração do Portal que irá permitir de uma forma simples e intuitiva atualizar todos os dados armazenados no BD. Para acessar esta área administrativa o usuário deverá informar no *browser* o endereço eletrônico do portal acrescentando “/Admin”<sup>32</sup> e efetuar o *login*. Os protótipos da tela de *login* e da tela principal de administração do *site* podem ser visualizados nas figuras 21 e 22 respectivamente.



The image shows a login form for the Intergenic DB Administration. At the top left is a DNA double helix icon. To its right, the text "Intergenic DB" is displayed in a large, bold, black font, with "DB" in a larger, gold-colored font. Below this, the word "Administration" is written in a smaller, bold, black font. Underneath, there are two input fields: "Email:" followed by a white rectangular box, and "Password:" followed by another white rectangular box. Below the password field is a button labeled "Logon" in a light gray box with a thin border.

Figura 21: *Login* de Administração



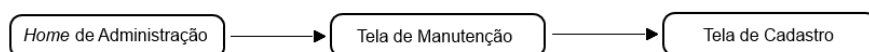
The image shows the home page of the Intergenic DB Administration. At the top left is a DNA double helix icon. To its right, the text "Intergenic DB" is displayed in a large, bold, black font, with "DB" in a larger, gold-colored font. Below this, the word "Administration" is written in a smaller, bold, black font. In the top right corner, the text "User: douglas.picolotto@gmail.com | LogOff" is displayed. Below the header, there are three main sections, each separated by a horizontal line. The first section is "Access Control" and contains two icons: "Groups" (two people) and "Users" (one person). The second section is "Molecular Biology" and contains five icons: "Genes" (DNA double helix), "Intergenic Regions" (DNA double helix with a red dot), "Organisms" (green leaf), "Predictor Methods" (blue globe), and "Roles" (DNA double helix with a red dot). The third section is "Others" and contains five icons: "Cities" (map), "Countries" (globe), "Import Data" (envelope with a blue arrow), "Publications" (document), and "States" (map).

Figura 22: *Home* de Administração

<sup>32</sup> Exemplo: <http://www.intergenicdb.com.br/Admin>

A *home* de administração do IntergenicDB será dividida em três grandes grupos com o intuito de facilitar a localização dos itens de *Menu*. O grupo *Access Control* (Controle de Acesso) possui os itens para manutenção da autenticação e autorização dos usuários do sistema. O grupo *Molecular Biology Information* (Informações sobre Biologia Molecular), possui os itens para manutenção dos cadastros utilizados nas consultas do sistema. O grupo *Other* (Outros) possui os itens para gerenciamento de artigos e publicações postados pelos usuários do sistema e também gerenciamento da fila de importação de arquivos de *upload*.

O modelo de navegação (Figura 23) dá área de administração é dividido em três níveis: *home* de administração (Figura 22), tela de manutenção e tela de cadastro. Desta forma, todos os itens de gerenciamento terão uma tela de manutenção e uma tela de cadastro.



**Figura 23: Modelo de navegação de telas da área de administração**

Ao clicar sobre um dos itens de gerenciamento na *home* de administração, o usuário será direcionado a tela de manutenção proposta pelo item (Figura 24). Nesta tela de manutenção, além de visualizar os dados armazenados, será possível efetuar as operações básicas de inserção, remoção, edição e pesquisa utilizando os botões , , e respectivamente. A ordenação dos dados será efetuada considerando a coluna mais relevante de cada tela de manutenção, no exemplo da Figura 24 será utilizada a coluna “*Name*”. Para retornar para a *home* de administração é necessário clicar sobre o botão .

Roles Maintenance			Search
Name			
Amino acid biosynthesis			
Biosynthesis of cofactors, prosthetic groups, and carriers			
Cell envelope			
Cellular processes			
Central intermediary metabolism			
Disrupted reading frame			
DNA metabolism			
Energy metabolism			
Fatty acid and phospholipid metabolism			
Hypothetical proteins			
Mobile and extrachromosomal element functions			
Pathogen responses			
Protein fate			
Protein synthesis			

**Figura 24: Exemplo de tela de manutenção**

Dois pontos importantes a serem observados sobre o funcionamento da tela de manutenção são:

- As operações de inserção (+) e edição (✎) dos itens serão efetuadas em uma tela de cadastro (Figura 25);
- Cada linha de resultado irá possuir uma operação de edição (✎) e outra de remoção (✖).

Os cadastros necessários são: Usuários, grupos de usuários, gene, região intergênica, organismo, método de predição, função (*role*), cidade, estado, país, publicação e fila de importação (*upload* de arquivos).

Para os campos que referenciam outras tabelas serão utilizados dois recursos para auxiliar na busca da informação desejada:

- *PickList*: O campo possuirá uma lupa (🔍) que ao receber o *click* do usuário irá abrirá uma tela de seleção em uma janela modal;
- *AutoCompleter*: Quando iniciada a digitação o campo irá apresentar, através de uma lista *Drop-Down*, as dados que iniciam com o valor digitado.

Ao finalizar a inclusão ou edição será necessário clicar no botão salvar (*Save*) para que os dados alterados sejam armazenados no banco de dados do IntergenicDB. Para sair sem salvar as alterações, será necessário clicar sobre o botão cancelar (*Cancel*) ou sobre o botão



The screenshot shows the 'Intergenic DB Administration' interface. At the top right, it displays 'User: douglas.picolotto@gmail.com | LogOff'. The main content area is divided into a table on the left and a form on the right.

**Intergenic Regions Table:**

Gene	Start	End	Sequence	Status	Tape 3	Tape 5	Actions
chromosomal replication protein DnaA	214	151	ATCTATTCATTATCTCAATCAGGCCGGGTTTGCITTT...	F	200	0	✎ ✖
bifunctional aspartokinase I/homoserine dehydrogenase I	274	0		F	274	0	✎ ✖
Chromosomal replication protein DnaA	318	0		F	318	0	✎ ✖
fused aspartokinase I and homoserine dehydrogenase I	336	256		F	336	256	✎ ✖
aspartokinase I, homoserine dehydrogenase I	336	256		F	336	256	✎ ✖
DNA polymerase III beta	1653	1572		F	1653	1572	✎ ✖
DNA polymerase III, beta	1867	1675		F	1867	1675	✎ ✖
conserved hypothetical protein similar to B. subtilis YaaA	3121	3014		F	3121	3014	✎ ✖
conserved hypothetical protein	4644	4466		F	4644	4466	✎ ✖
conserved hypothetical protein	5213	5001		F	5213	5001	✎ ✖
conserved hypothetical protein	5682	5531		R	5682	5531	✎ ✖

**New Intergenic Region Form:**

Gene: (alpha)-a  
 Sequence: (alpha)-aspartyl dipeptidase  
 Tape 3:   
 Tape 5:   
 Length:   
 Status: Forward

Buttons: Save, Cancel

**Figura 25: Exemplo de tela de cadastro (Inclusão)**

### 5.1.2 Autenticação e autorização de usuários

A primeira versão do IntergenicDB possui um sistema de autenticação de usuários simples, utilizado apenas para controlar o acesso à funcionalidade de *upload* de arquivos. Para a nova versão será necessária uma forma mais elaborada de tratar os usuários do sistema, visto que existirão dois tipos de usuário:

- *User*: Usuários com acessos mínimos, sem controles de autorização, que poderão submeter arquivos através da ferramenta de *upload* (são os usuários existentes atualmente).
- *Admin*: Usuários que terão acesso a área de administração do portal, e poderão modificar os dados existentes de acordo com o nível de permissão concedido.

Quando o cadastro do usuário for efetuado pela ferramenta existente (Figura 26), o mesmo será sempre armazenado como um usuário do tipo *user*. Para cadastrar novos usuários administradores (*admin*) será necessário utilizar a área de administração do IntergenicDB, e o cadastro deverá ser efetuado por um usuário *admin* existente com as devidas permissões.

Para gerenciar a governança entre os usuários do sistema, será utilizado o cadastro de Grupos de Usuários (*User Groups*), que irá permitir criar e modificar grupos e suas permissões. O cadastro de grupos de usuários pode ser visualizado na figura 27.



Figura 26: Cadastro de usuários do tipo *user*

**Edit Group**

Name: Administradores

Menu	View	Create	Edit	Delete
Access Control	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Groups	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Users	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Molecular Biology	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Genes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Intergenic Regions	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Organisms	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Predictor Methods	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Roles	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Others	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Cities	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Countries	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Import Data	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Publications	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
States	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Save Cancel

**Figura 27: Cadastro de grupos de usuários**

No cadastro de usuários, poderá ser definido o tipo do usuário a ser criado. Se o usuário for criado como um *user* (Figura 28) será solicitado apenas o nome, sobrenome, o *e-mail* e a senha. Quando for criado como *admin* (Figura 29) será solicitado também o grupo de usuários ao qual ele pertence e um telefone para contato.

**Edit User**

Type: User

Active:

Name: Douglas

Last Name: Picolotto

Email: douglas.picolotto@gmail.com

Password: .....

Confirm Password: .....

Save Cancel

**Figura 28: Cadastro de usuário básico (*user*)**

Type	Administrator
Active	<input checked="" type="checkbox"/>
Name	Douglas
Last Name	Picolotto
Group	Administradores
Phone Number	11223311
Email	douglas.picolotto@gmail.com
Password	.....
Confirm Password	.....

**Figura 29: Cadastro de usuário avançado (admin)**

Usuários visitantes que se cadastram no sistema através da página inicial, irão receber no *e-mail* informado uma solicitação de confirmação de *e-mail*. Somente após efetuar esta confirmação é que os usuários terão seu *status* alterado para ativo.

### 5.1.3 Upload de arquivos

Atualmente o IntergenicDB possibilita que o usuário efetue *upload* de arquivos mediante autenticação. A interface é simples, após o *login* é disponibilizada uma nova aba “*Upload*”, que possui apenas um campo para selecionar o arquivo a ser carregado (Figura 30).

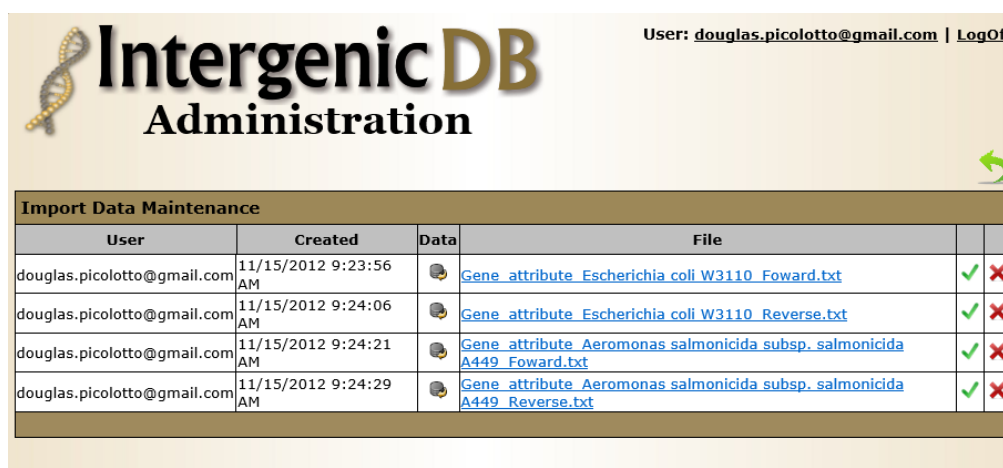


**Figura 30: Upload de arquivos**

Na solução desenvolvida, o *upload* de arquivos deposita os dados importados diretamente na base de dados, isto pode comprometer a integridade e veracidade destes dados. Para evitar o mau uso do portal, é necessário que um usuário responsável pelo sistema analise estes dados antes de importá-los. Como solução para este problema, o processo de *upload* de arquivos será dividido em quatro etapas:

- *Upload*: Etapa onde o usuário envia o arquivo a ser importado para análise. Este arquivo é um arquivo texto que possui um formato específico do IntergenicDB, onde o separador de dados utilizado é o caractere de tabulação. Este arquivo existe em duas formas diferentes definidas pela orientação da fita de DNA. Por este motivo, para efetuar a importação completa de um organismo, região intergênica e gene é necessário importar dois arquivos diferentes, um com as informações no sentido “*forward*” e outro com as informações no sentido “*reverse*” (DAVANZO, 2010). O exemplo do formato do arquivo pode ser visualizado no ANEXO B.
- *Análise*: Etapa onde um usuário responsável analisa os dados importados;
- *Importação*: Etapa onde o usuário responsável submete os dados importados ao banco de dados através da tela de administração do IntergenicDB.
- *Notificação*: Etapa onde o sistema notifica o usuário que submeteu o arquivo de que a importação foi concluída.

Na tela de Manutenção da Fila de Importação (*Import Queue*) (Figura 31), acessada pela *Home* de Administração, o usuário administrador poderá verificar quem efetuou a importação, a data da importação, os dados que serão importados e também o arquivo utilizado para a importação. O usuário poderá também aprovar ou excluir a importação. Quando os dados forem aprovados e importados, o usuário que efetuou o *upload* será notificado por *e-mail* da conclusão da importação.



The screenshot shows the 'IntergenicDB Administration' interface. At the top right, it displays 'User: douglas.picolotto@gmail.com | LogOff'. Below the header is a table titled 'Import Data Maintenance' with the following columns: User, Created, Data, File, and two empty columns for status. The table contains four rows of import data.

User	Created	Data	File		
douglas.picolotto@gmail.com	11/15/2012 9:23:56 AM		<a href="#">Gene_attribute_Escherichia_coli_W3110_Foward.txt</a>	✓	✗
douglas.picolotto@gmail.com	11/15/2012 9:24:06 AM		<a href="#">Gene_attribute_Escherichia_coli_W3110_Reverse.txt</a>	✓	✗
douglas.picolotto@gmail.com	11/15/2012 9:24:21 AM		<a href="#">Gene_attribute_Aeromonas_salmonicida_subsp_salmonicida_A449_Foward.txt</a>	✓	✗
douglas.picolotto@gmail.com	11/15/2012 9:24:29 AM		<a href="#">Gene_attribute_Aeromonas_salmonicida_subsp_salmonicida_A449_Reverse.txt</a>	✓	✗

**Figura 31: Manutenção da fila de importação**

Para garantir que o processamento dos arquivos importados esteja sendo executado corretamente, serão criados arquivos de teste de importação e submetidos ao portal.

#### 5.1.4 Carga de dados

Para possibilitar a utilização do portal, é necessário carregar o banco de dados do mesmo com dados iniciais, afim de que as consultas efetuadas retornem resultados válidos. Para isto, como requisitos foram definidas duas cargas iniciais necessárias para o IntergenicDB: a primeira é necessária para suprir informações sobre organismos, genes e regiões intergênicas; e a segunda é necessária para carregar os métodos de predição de promotores para estas regiões intergênicas.

Para carregar as regiões intergênicas, serão utilizados os arquivos no formato de *upload* do IntergenicDB providos pela Profa. Dra. Scheila de Ávila e Silva.

As informações de predição<sup>33</sup> serão extraídas do serviço CGI<sup>34</sup> da ferramenta *web Prompredict*<sup>35</sup>. O *Prompredict* é um servidor *web* desenvolvido pela *Molecular Biophysics Unit*<sup>36</sup> no IISC<sup>37</sup>, que é utilizado para a identificação de promotores em sequências de DNA, através da estabilidade energética destas sequências.

A consulta será executada utilizando uma requisição para a URL <http://nucleix.mbu.iisc.ernet.in/cgi/bin/prompredict/prompredict.cgi>. No corpo da requisição devem ser enviados os seguintes parâmetros: a sequência genômica em formato *fasta*; o tamanho da janela a ser considerada para a predição; se é para considerar o %GC da sequência ou de todo o genoma; e se a pesquisa deve ser efetuada somente na direção da transcrição ou em ambas as direções. A ferramenta irá retornar o conteúdo em formato HTML. Para recuperar este conteúdo em formato de texto, em seguida será gerada uma segunda requisição utilizando a URL [http://nucleix.mbu.iisc.ernet.in/cgi-bin/prompredict/download.cgi?ID=/var/tmp/prompredict/prompredict\\_results.txt](http://nucleix.mbu.iisc.ernet.in/cgi-bin/prompredict/download.cgi?ID=/var/tmp/prompredict/prompredict_results.txt). Dentre as informações retornadas estão o identificador da sequência, o comprimento da sequência informada, o %GC da sequência, o início e fim da sequência promotora, o comprimento da sequência promotora e a sequência promotora.

---

<sup>33</sup> Prognóstico

<sup>34</sup> CGI: *Common Gateway Interface*

<sup>35</sup> <http://nucleix.mbu.iisc.ernet.in/prompredict/prompredict.html>

<sup>36</sup> <http://mbu.iisc.ernet.in/>

<sup>37</sup> IISC: *Indian Institute of Science*



### 5.1.5 Área de postagem

Um dos requisitos é a criação de uma área de postagem no IntergenicDB onde possam ser visualizados documentos relacionados a promotores. Para isso, será criada uma nova aba na página principal do portal chamada “News” (Figura 32).

A nova aba irá exibir as publicações existentes em ordem decrescente por data de criação, e paginadas, limitando em quinze publicações por página para garantir um bom desempenho na recuperação das publicações e evitar uma rolagem muito extensa na área disponível. Irá existir também um campo para consulta, que permitirá filtrar as publicações por: método de predição, ano e mês de publicação, título, autor e instituto. Ao clicar sobre a publicação desejada, a mesma será exibida em uma nova aba no *browser*.

A manutenção dos dados da área de postagem ocorrerá através do item de gerenciamento “Publication” no grupo “Other” da *Home* de Administração. Neste grupo, será possível também dar manutenção as novas entidades “City”, “State” e “Country” que possuem relacionamento com “Publication”. Nele poderão ser incluídos, excluídos ou editados os *posts* desejados.

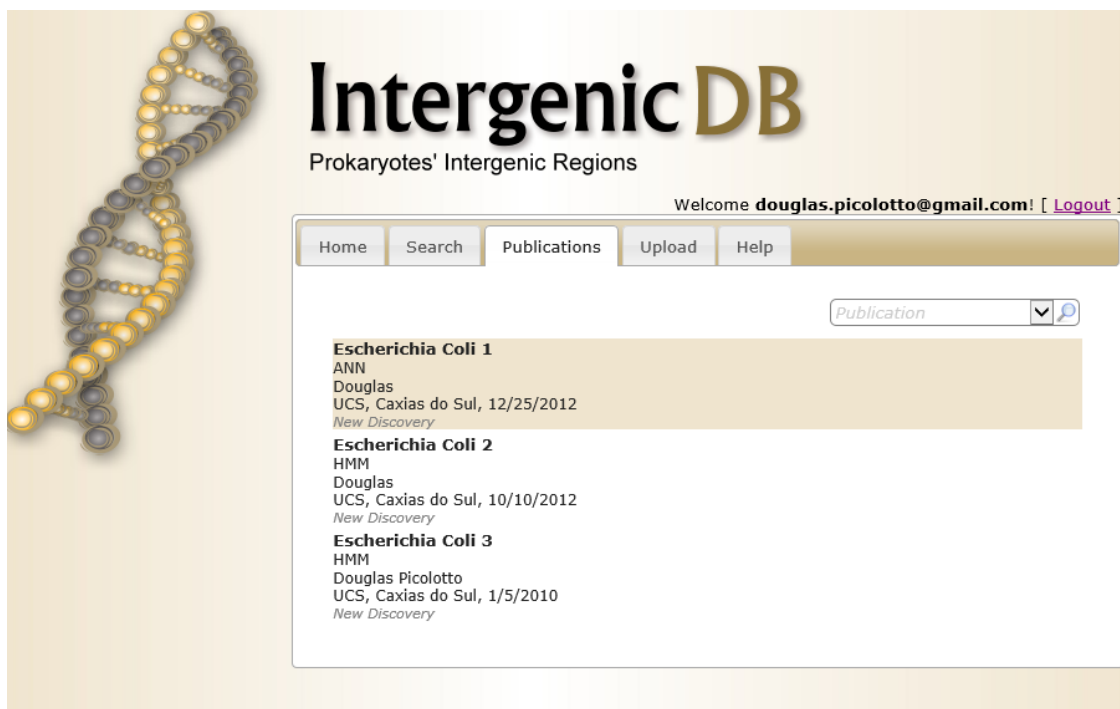


Figura 32: Área de postagem

### 5.1.6 Internacionalização

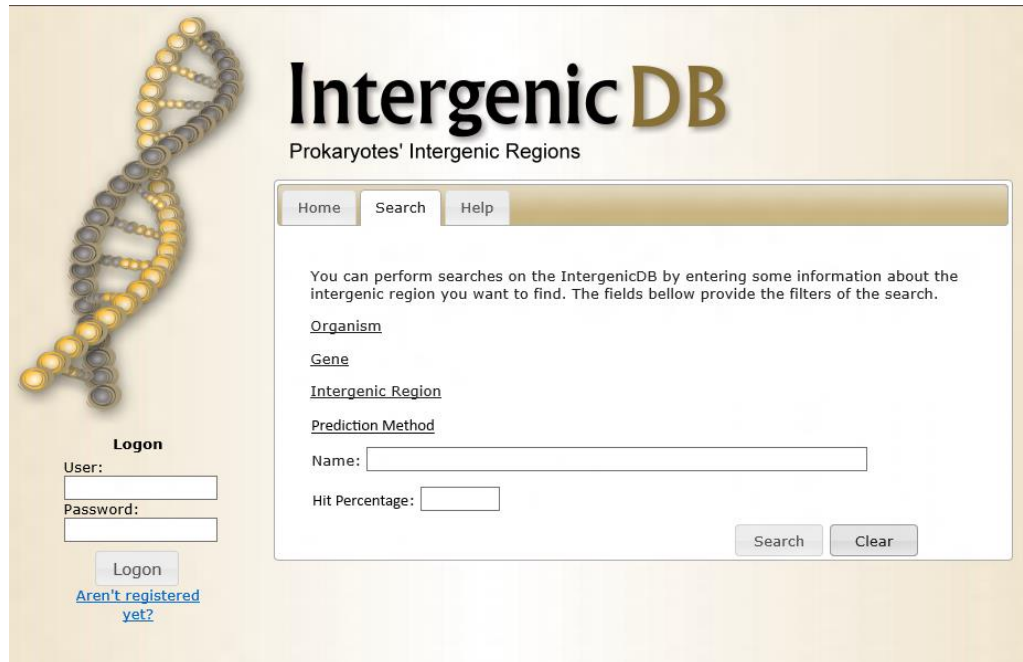
A língua nativa utilizada para todas as mensagens e textos do IntergenicDB é o Inglês. Contudo, por se tratar de um portal desenvolvido na Universidade de Caxias do Sul, é

importante que o mesmo tenha suporte para o Português. Para dar este suporte será explorado o cabeçalho HTTP<sup>38</sup> “Accept-Language” que é utilizado para especificar a linguagem utilizada pelo usuário (W3C, 2012).

O cabeçalho HTTP é enviado pelo navegador de internet do computador que efetua a requisição em um formato semelhante à “Accept-Language: pt-BR,pt;q=0.8,en-US;q=0.5,en;q=0.3”. As línguas no cabeçalho são separadas por vírgulas (,), onde a primeira é a principal e as demais são línguas preferenciais tendo seu grau de importância definido pelo valor após o ponto e vírgula (;).

### 5.1.7 Consultas e testes

Atualmente o IntergenicDB permite efetuar consultas utilizando como elementos de pesquisa: organismos, genes e regiões intergênicas (Figura 19). É possível gerar consultas utilizando os elementos individualmente ou combinando os mesmos gerando consultas mais precisas. Como um dos propósitos do portal são os métodos de predição de promotores, além destes elementos será incluída também a possibilidade de consultar através do método de predição (Figura 33).



**Figura 33: Tela de pesquisa**

Para validar as alterações efetuadas e as consultas existentes do portal, serão criados os seguintes cenários de consulta:

<sup>38</sup> HTTP: *Hypertext Transfer Protocol*

- *Consultas por elemento:* organismo, gene, região intergênica e método de predição;
- *Consultas combinadas:* organismo x genes, gene x região intergênica e método de predição x genes.

Os resultados obtidos em cada cenário serão tabelados, onde serão registrados os tempos de resposta do portal, a quantidade de resultados obtidos e se os resultados obtidos são consistentes.

## 5.2 MODELO CONCEITUAL (ER)

Após definidos os requisitos e propostas as soluções para os mesmos, foi possível identificar as alterações necessárias que deverão ser efetuadas no modelo ER proposto pela aluna Vanessa (DAZANZO, 2010). O novo modelo ER está representado na figura 34.

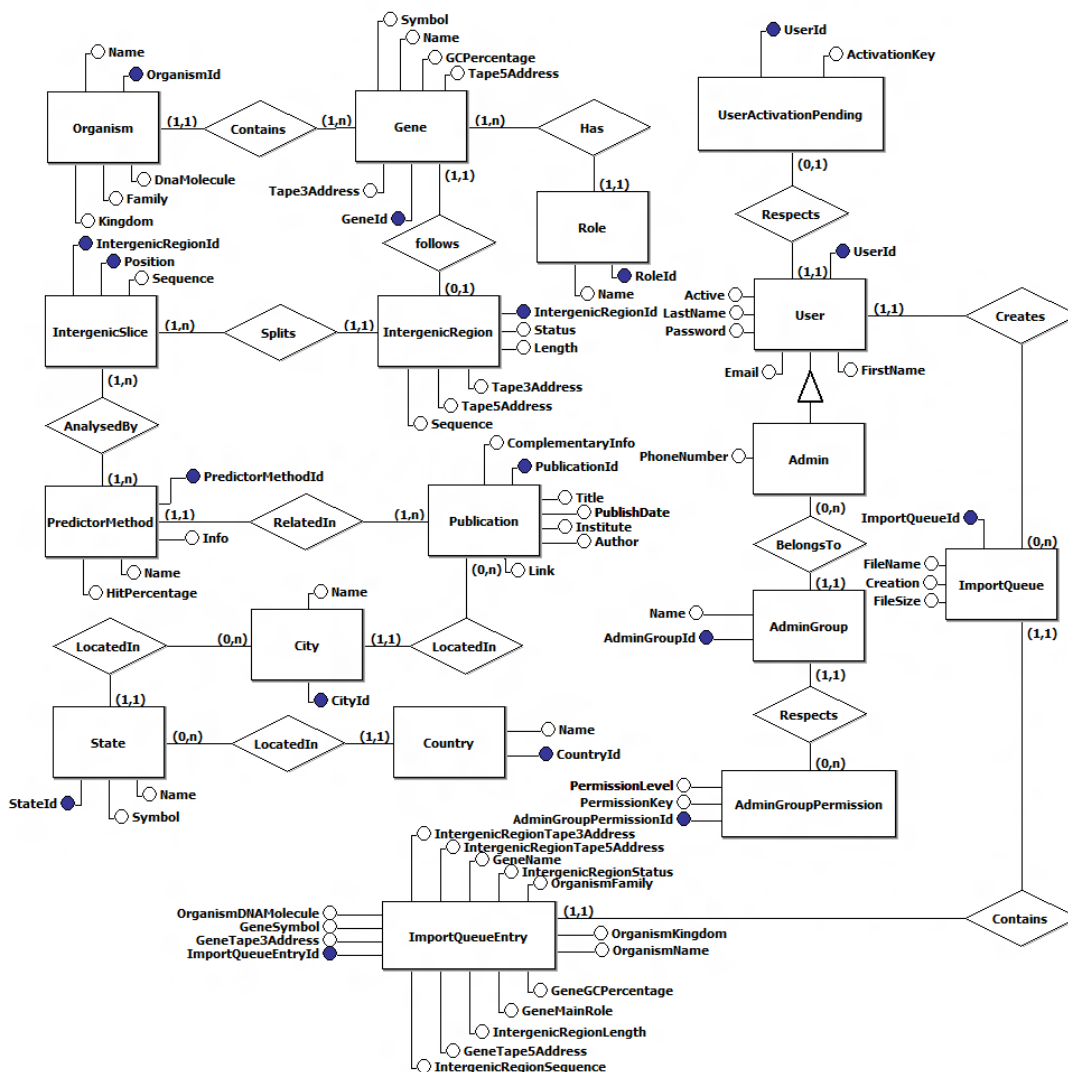



Figura 34: Modelo conceitual IntergenicDB 2.0

As principais alterações efetuadas no modelo ER são:

- Remoção dos atributos *City* e *Country* da entidade *Publication*, e criação das entidades *City*, *State* e *Country* com seus devidos relacionamentos. Esta alteração é necessária para garantir a integridade dos dados relacionados a localização da publicação;
- Alteração da entidade *ImportQueue* para armazenar os dados temporários do arquivo de *upload*, do usuário que efetuou a importação e da data da importação. Isto é necessário para efetuar a análise do arquivo após o *upload*;
- Criação da entidade *ImportQueueEntry* para armazenar os dados temporários do arquivo de *upload* processado. Esta entidade é necessária para evitar o reprocessamento do arquivo quando o usuário desejar verificar os mesmos na análise da fila de importação (Figura 31), opção .
- Criação da entidade *UserActivationPending* que será utilizada para ativar os usuários que se registram no portal através da página principal do IntergenicDB;
- Criação da entidade *Admin* (especialização de *User*) para tratar usuários administradores do sistema;
- Criação da entidade *AdminGroup* que irá armazenar informações sobre grupos de administradores com permissões diferenciadas;
- Criação da entidade *AdminGroupPermission* que irá armazenar as informações de permissões dos grupos de administradores (*AdminGroup*).

Utilizando este modelo conceitual, foi criado o modelo lógico apresentado na figura

35.

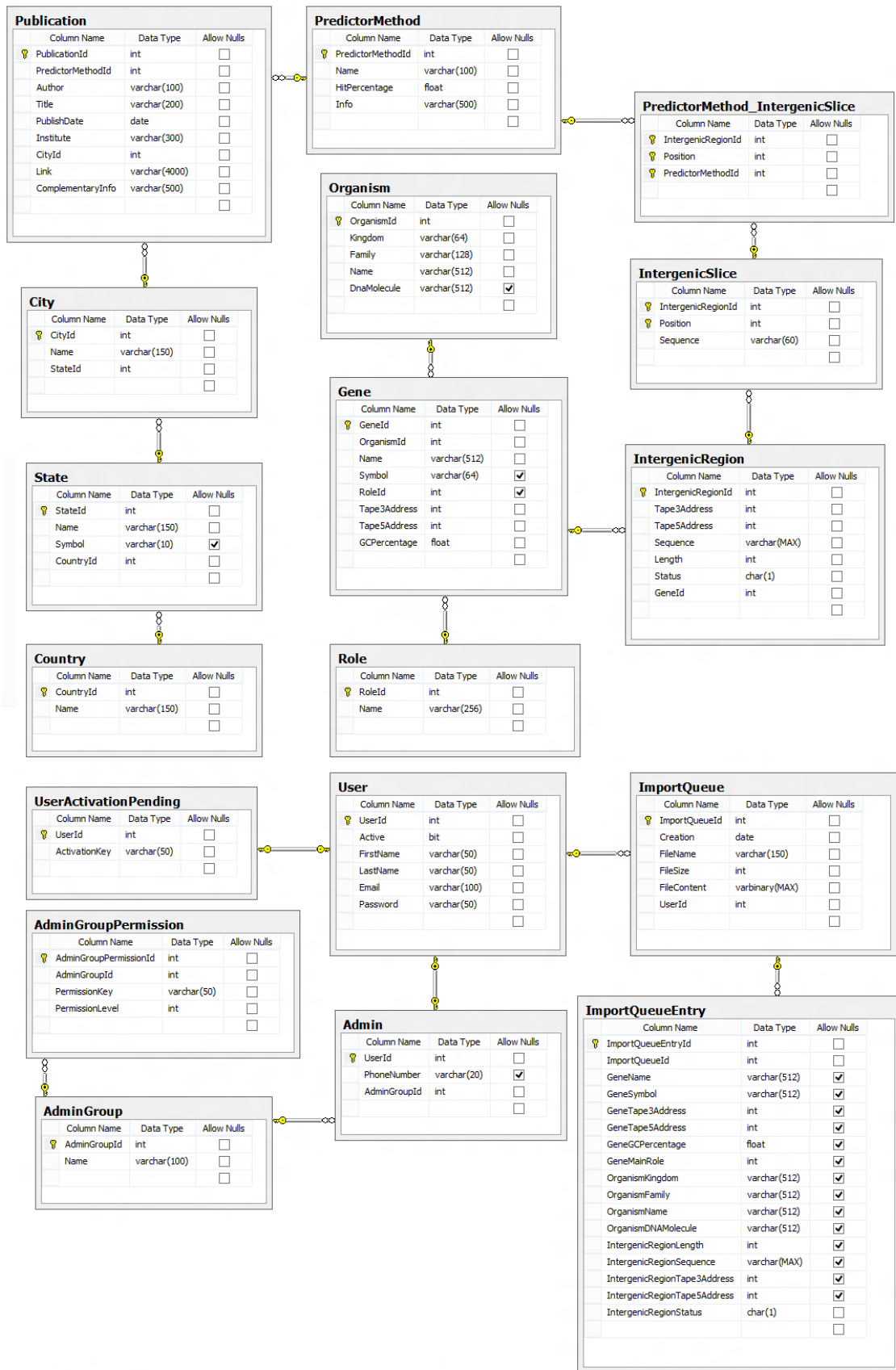


Figura 35: Modelo lógico IntergenicDB 2.0

### 5.3 DIAGRAMAS DE CLASSES

Nesta seção estão definidos os diagramas de classes que serão utilizados para desenvolver as soluções apresentadas para a nova versão do IntergenicDB. Os diagramas de classes completos ou que não serão modificados neste momento estão disponíveis nos anexos C, D e E deste trabalho.

Para o primeiro requisito será criado um novo *controller*<sup>39</sup> chamado *AdministrationController* que terá a funcionalidade de apresentar a tela de administração do IntergenicDB e tratar as respostas e ações efetuadas na mesma (Figura 36). Ele irá solicitar ao serviço *MenuProvider* os *MenuItems* disponíveis, que por sua vez irá carregar os mesmos de um arquivo de configuração. Além disso, serão criadas todas as *views* necessárias.

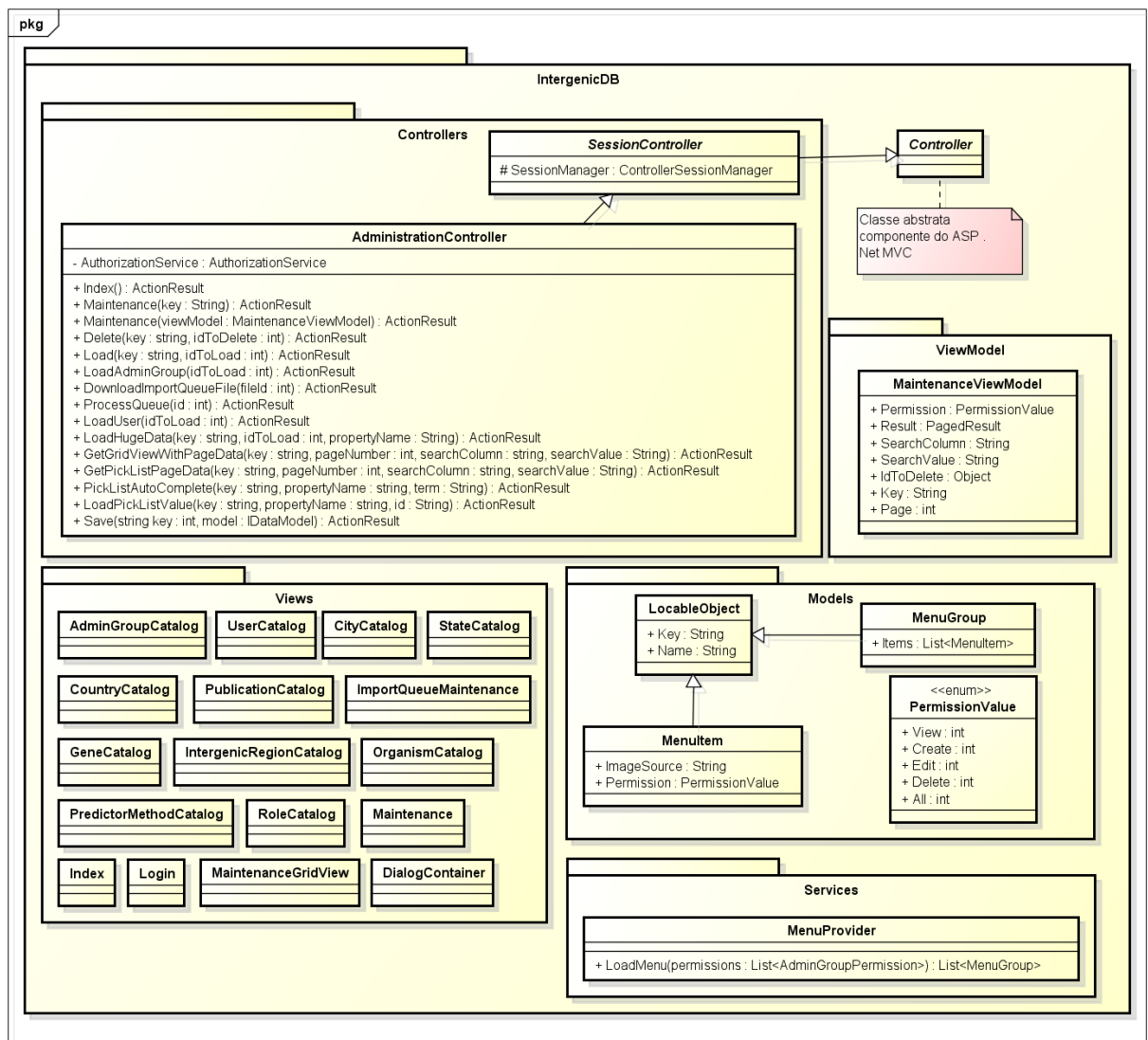


Figura 36: Diagrama de classes do requisito 1

<sup>39</sup> *Controller*: Componente do design pattern MVC

No segundo requisito será criado um serviço de autorização chamado *ControllerSessionManager* (Figura 37) que irá prover uma série de funcionalidades relacionadas ao usuário: como identificar se o usuário autenticado é administrador, ou se o mesmo possui as devidas permissões para executar uma operação específica.

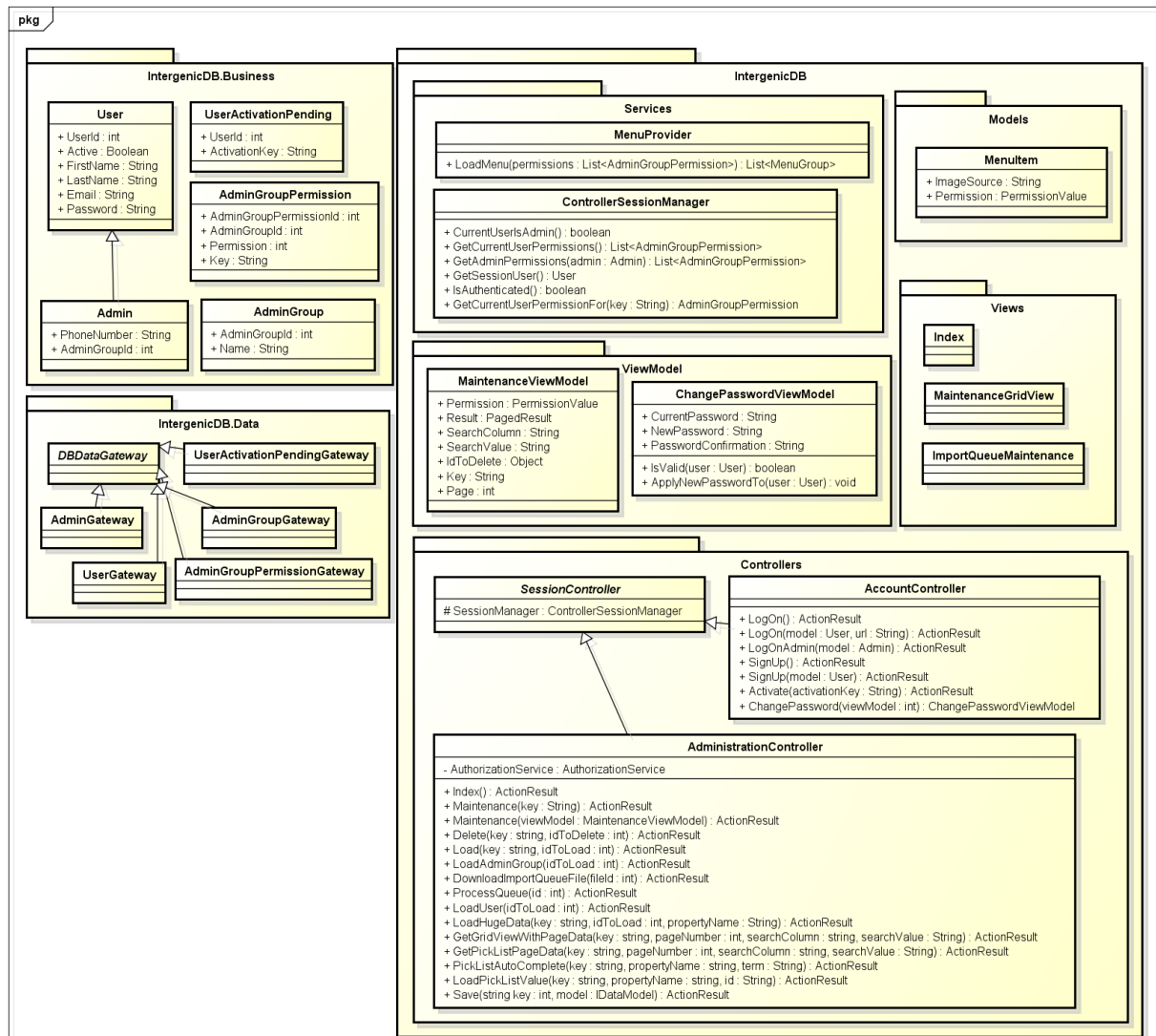


Figura 37: Diagrama de classes do requisito 2

O terceiro requisito (*upload* de arquivos) já está funcionando na versão atual do IntergenicDB. As alterações necessárias a serem feitas são: remoção da carga automática dos arquivos na classe *IntergenicDB.Business.IntergenicDatabaseService*; criação e/ou alteração das classes de modelo necessárias; e criação da *view* para auditoria (*ImportQueueMaintenance*) dos arquivos de *upload* (Figura 38).



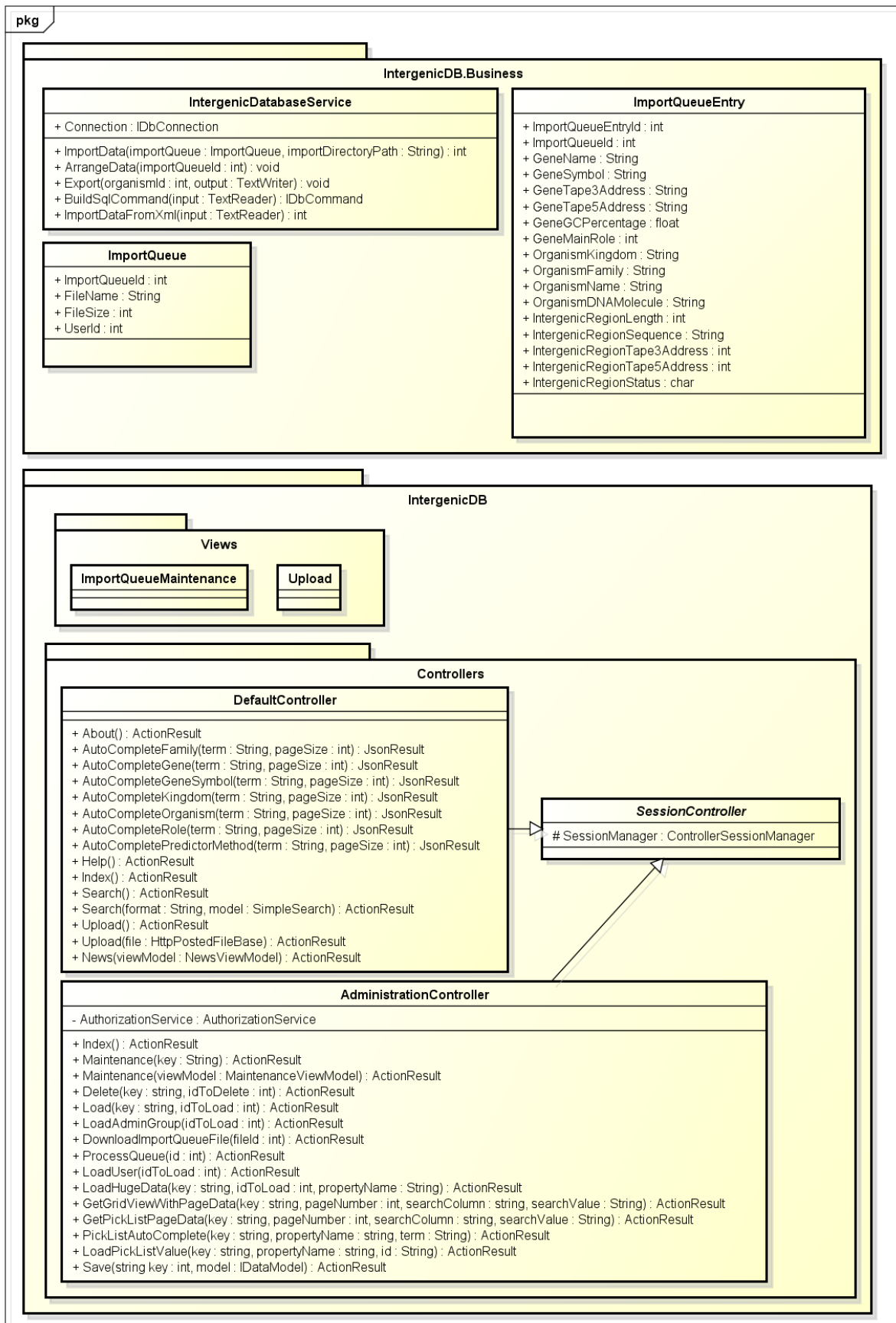
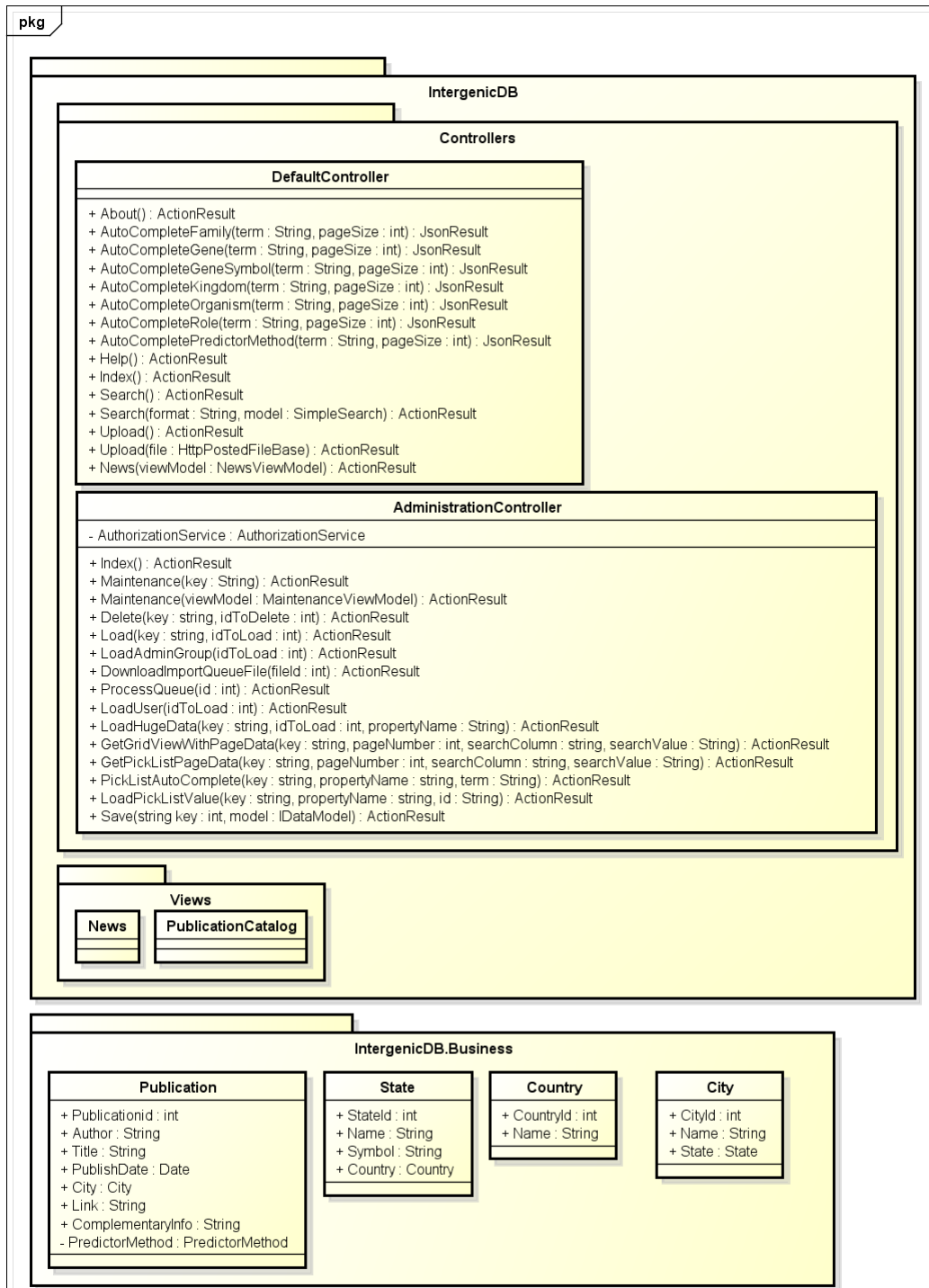


Figura 38: Diagrama de classes do requisito 3



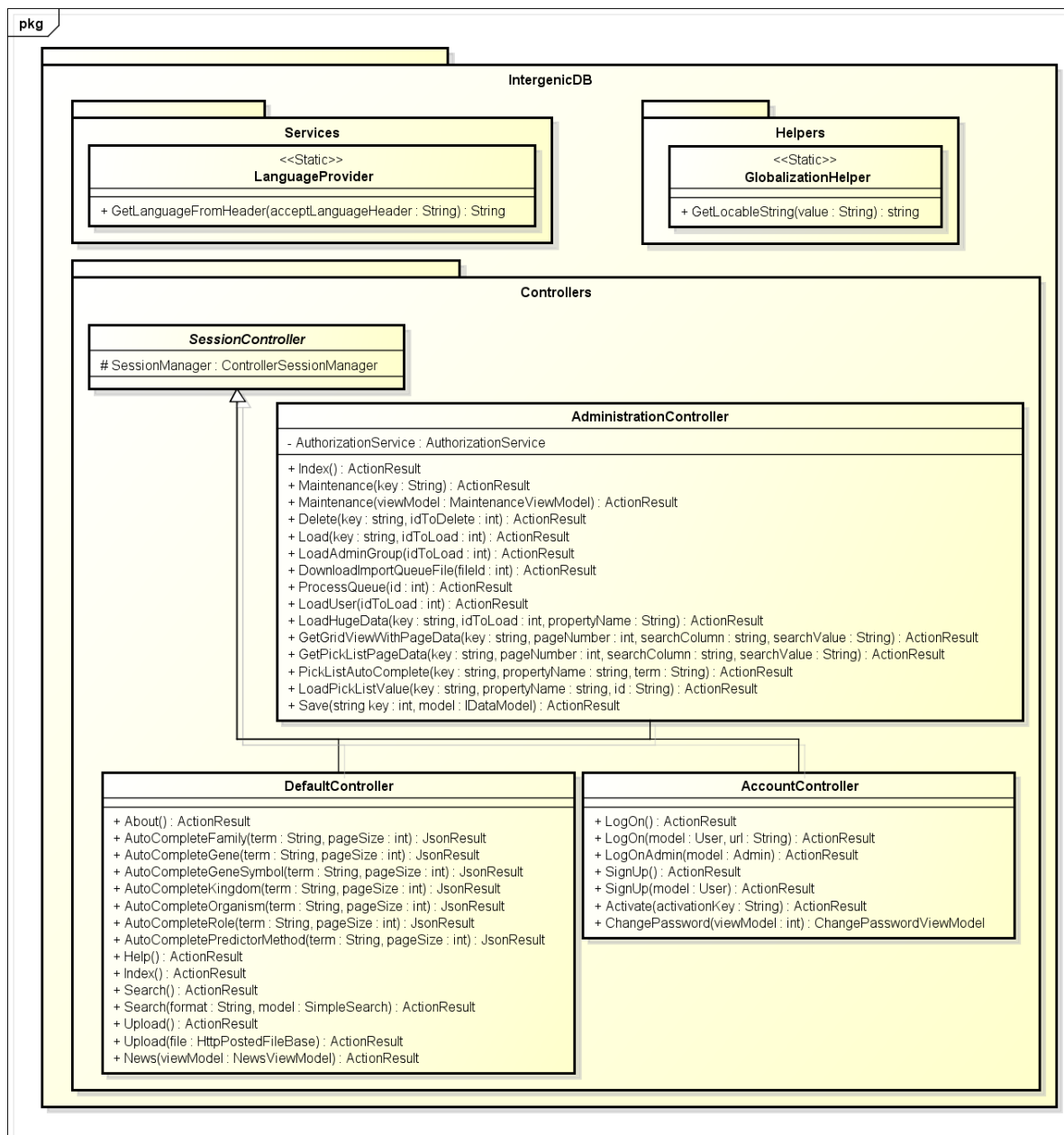
A área de postagem (quinto requisito) será apresentada utilizando o *controller* principal (*DefaultController*) existente hoje no IntergenicDB (Figura 39). Foi adicionado o método *News* que irá processar a requisição recebendo como parâmetros um objeto do tipo *NewsViewModel* que armazenará o tamanho da página, o número da página e o termo para pesquisa.



powered by Astah

**Figura 39: Diagrama de classes do requisito 5**

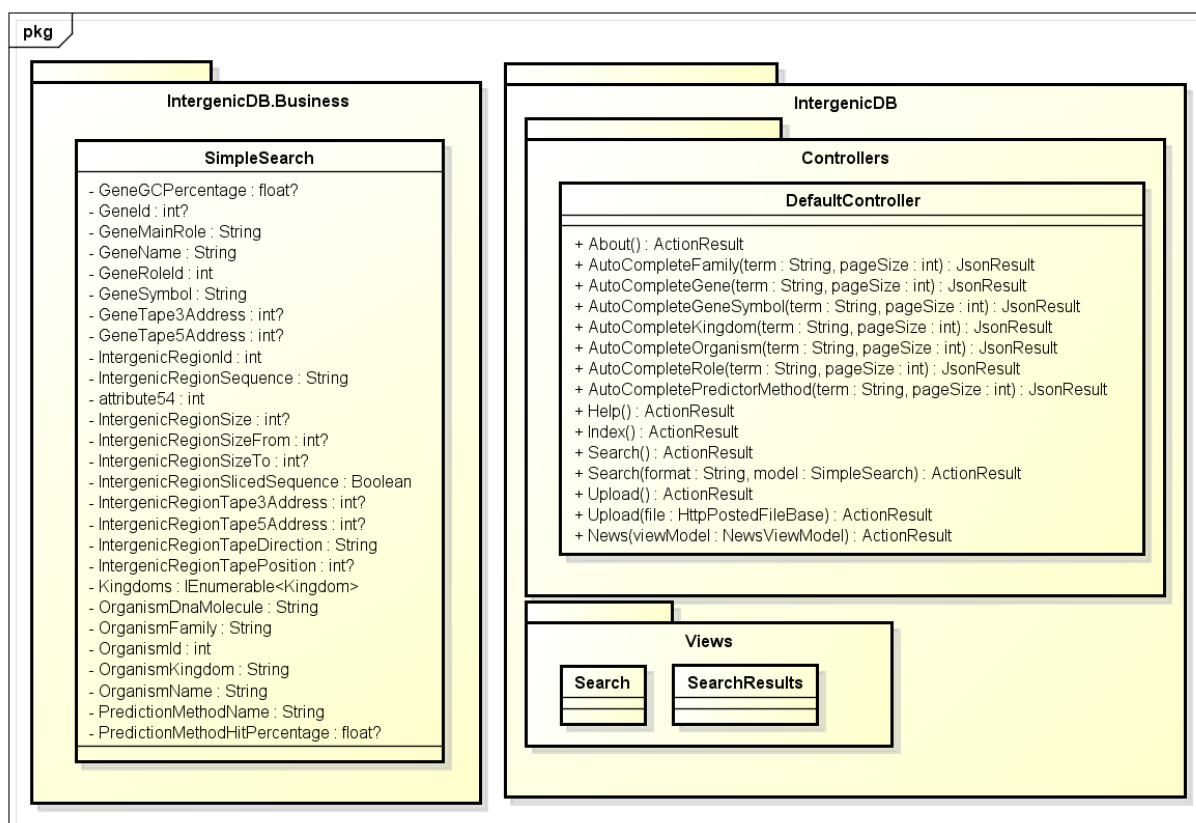
Para tratar a internacionalização do sistema (sexto requisito), será criado um serviço chamado *LanguageProvider* que irá prover o idioma armazenado no cabeçalho *accept-language*. Este idioma será utilizado pelo *SessionController* para definir a linguagem da sessão do ASP.Net. Para recuperar as descrições e textos traduzidos no idioma utilizado pelo sistema, será criada uma classe utilitária chamada *GlobalizationHelper* (Figura 40).



powered by Astah

**Figura 40: Diagrama de classes do requisito 6**

No sétimo e último requisito, consultas e testes, foi incluída uma nova opção de pesquisa por métodos de predição. Para tratar esta nova pesquisa, será alterada apenas a classe *SimpleSearch*, que possui todas as propriedades de pesquisa definidas, a classe *DefaultController* (Figura 41), e a *view* responsável pela apresentação da opção de busca.



powered by Astah

**Figura 41: Diagrama de classes do requisito 7**

## 5.4 FERRAMENTAS

Nesta seção serão apresentadas as ferramentas utilizadas para o desenvolvimento das soluções propostas neste capítulo.

### 5.4.1 Visual Studio 2010

A ferramenta utilizada para o desenvolvimento é o *Microsoft Visual Studio 2010*<sup>40</sup> (Figura 42), que é a plataforma de desenvolvimento da *Microsoft*. Sendo uma das mais completas ferramentas disponíveis no mercado, possui suporte a diversas linguagens de programação possibilitando o desenvolvimento de aplicações de console, desktop e *web*.

<sup>40</sup> <http://www.microsoft.com/visualstudio/en-us>

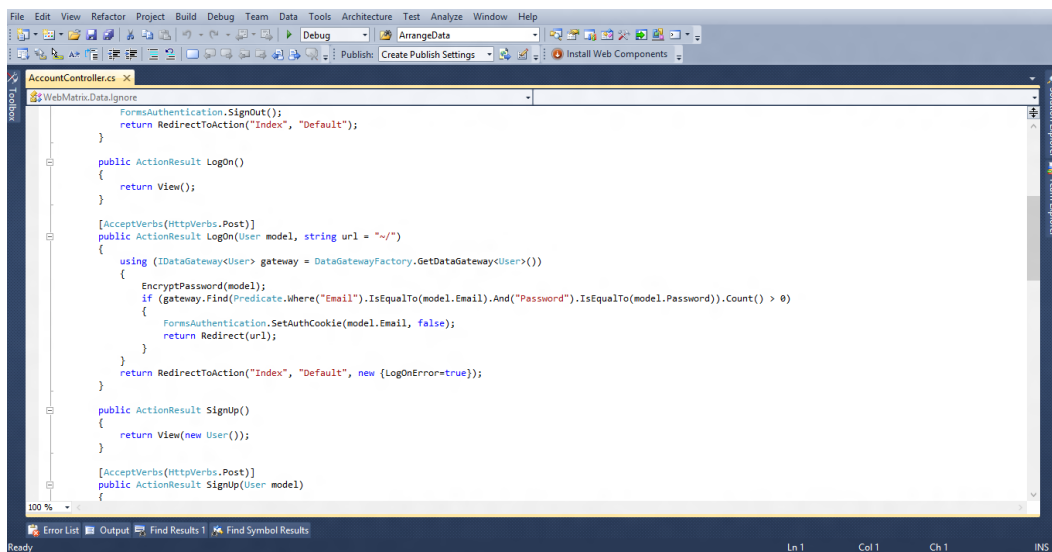


Figura 42: Microsoft Visual Studio 2010

## 5.4.2 Paint .NET

Para o tratamento de imagens, será utilizado o *Paint .NET*<sup>41</sup> (Figura 43), que é um software para edição de imagens *freeware*<sup>42</sup>. Desenvolvido para a plataforma Windows, surgiu como um projeto de graduação onde o objetivo era criar uma ferramenta de edição de imagens grátis que pudesse substituir o *Microsoft Paint*<sup>43</sup>. Possui uma série de ferramentas poderosas, efeitos, e suporte a camadas o que o tornam uma excelente ferramenta para este propósito.

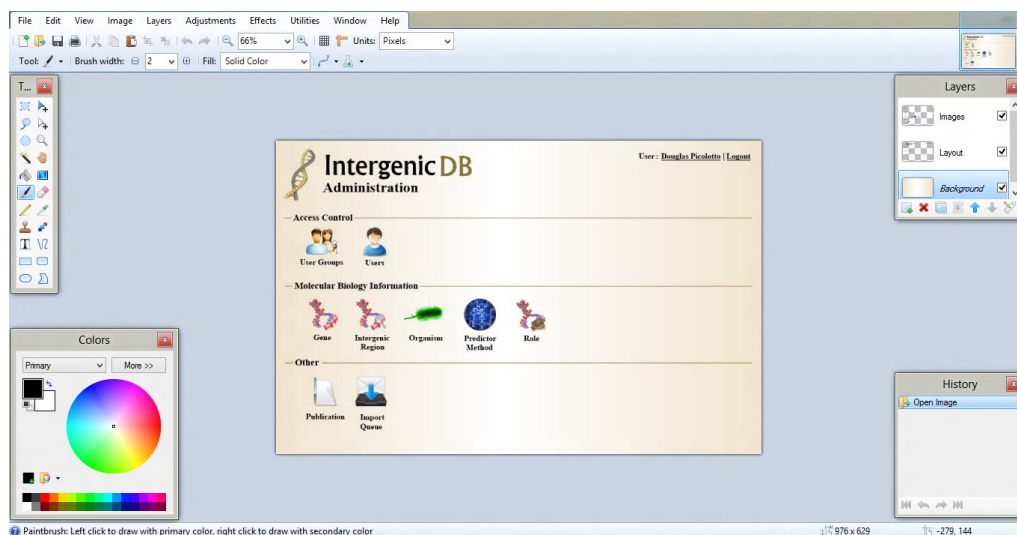


Figura 43: Paint .NET

<sup>41</sup> <http://www.getpaint.net/>

<sup>42</sup> *Freeware*: Tipo de *software* disponibilizado sem custo ou mantido através de doações.

<sup>43</sup> <http://windows.microsoft.com/en-us/windows7/products/features/paint>

### 5.4.3 MySQL Query Browser

O MySQL *Query Browser*<sup>44</sup> (Figura 44) é uma ferramenta distribuída dentro do pacote de instalação do MySQL *Administrator*<sup>45</sup> que permite efetuar o gerenciamento do SGBD de uma forma prática e rápida através de uma interface gráfica, bastante intuitiva, e possibilitando operações DML<sup>46</sup> e DDL<sup>47</sup>.

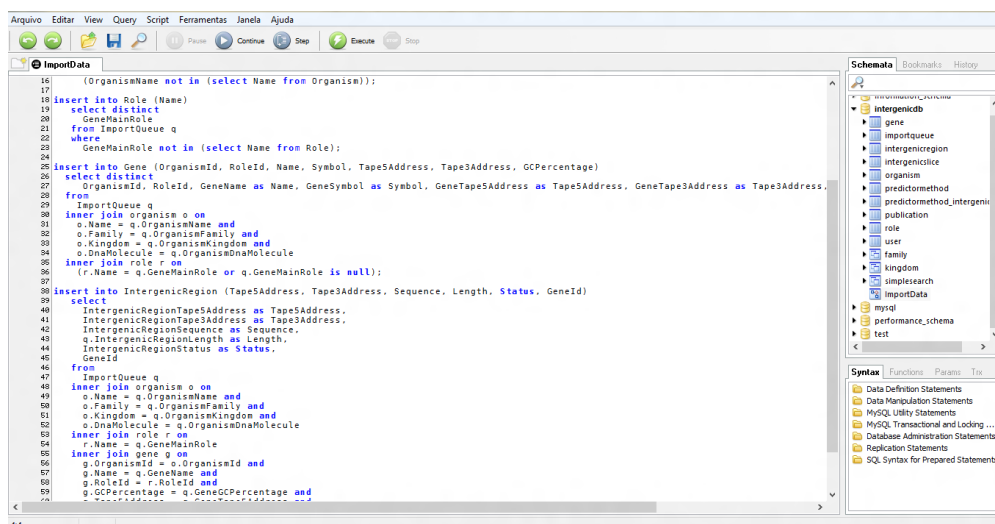


Figura 44: MySQL Query Browser

## 5.5 CONSIDERAÇÕES FINAIS

Os novos requisitos definidos para a nova versão do IntergenicDB irão prover uma série de funcionalidades para auxiliar os usuários do sistema no gerenciamento do portal, e para prover novas informações e opções de consulta para visitantes. Diversas alterações na versão atual serão necessárias, porém o seu desenvolvimento é de suma importância para tornar o portal operacional.

No próximo capítulo, será efetuado o desenvolvimento destes requisitos e descritos os passos efetuados durante este processo.

<sup>44</sup> <http://dev.mysql.com/doc/query-browser/en/>

<sup>45</sup> <http://dev.mysql.com/doc/administrator/en/>

<sup>46</sup> DML: Data Manipulation Language

<sup>47</sup> DDL: Data Definition Language

## 6 DESENVOLVIMENTO

Neste capítulo serão apresentados os principais pontos do desenvolvimento de cada um dos requisitos apresentados no capítulo 5 (seção 5.1), e também algumas das dificuldades encontradas durante este processo.

### 6.1 MANUTENÇÃO DOS DADOS

Por ser o maior requisito proposto neste trabalho, o desenvolvimento do mesmo foi dividido em quatro etapas menores que serão apresentadas nas próximas seções.

#### 6.1.1 Login

A interface de *login* de administração manteve os conceitos definidos na etapa de projeto deste trabalho. A principal modificação efetuada foi a utilização do *controller*<sup>48</sup> *AccountController* (Figura 45) ao invés do *AdministrationController* para efetuar as operações relacionadas ao *login* de administração. O motivo desta alteração é apenas conceitual, com o intuito de manter as operações desta natureza em um único *controller*.

```
[AcceptVerbs(HttpVerbs.Post)]
(A) public ActionResult LogOnAdmin(Admin model)
{
    using (var gateway = DataGatewayFactory.GetDataGateway<Admin>())
    {
        (B) model.Password = CryptographyHelper.EncodePassword(model.Password);
        (C) if (gateway.Find(Predicate.Where("Email").IsEqualTo(model.Email).And("Password")
            |.IsEqualTo(model.Password).And("Active").IsEqualTo(true)).Count() > 0)
        {
            (D) FormsAuthentication.SetAuthCookie(model.Email, false);
            return RedirectToAction("Index", "Admin");
        }
    }
    (E) return RedirectToAction("Index", "Admin", new { LogOnError = true });
}
```

Figura 45: Detalhes de implementação no *AccountController*

A principal operação incluída no *AccountController* é chamada *LogOnAdmin* que tem como funcionalidade principal autenticar um usuário administrador no sistema. A mesma recebe as informações do *login* através do argumento “*model*” (A), efetua a criptografia do *password* do mesmo (B), e executa uma pesquisa no banco de dados utilizando o *LP.Framework* (DAVANZO,2010). Esta pesquisa consiste em localizar ao menos um

<sup>48</sup> *Controller*: Elemento conceitual do padrão MVC (*Model-View-Controller*)

administrador ativo, que possua o *e-mail* e *password* informado (C). Se o resultado da pesquisa for bem sucedido, será armazenado o *e-mail* do administrador como *cookie*<sup>49</sup> de autorização no navegador e o usuário poderá acessar a área de administração do sistema (D), caso contrário, será retornado ao usuário que as informações de *login* são inválidas (E).

### 6.1.2 Administração

A tela principal da administração do sistema foi criada utilizando uma classe de serviço desenvolvida para criação de menus utilizando um arquivo de configuração no formato XML<sup>50</sup> chamada *MenuProvider*. Os principais elementos deste arquivo de configuração são o nodo *MenuGroup* (A), que define um grupo de menus dentro da área de administração, e o nodo *MenuItem* (B) que representa os itens de menu dentro de um grupo. Este arquivo de configuração pode ser visualizado na figura 46.

```
<?xml version="1.0" encoding="utf-8" ?>
<Menu>
  (A) <MenuGroup key="AccessControl">
    (B) <MenuItem key="AdminGroup" imageSource="../../content/images/Menu/AccessGroup128x128.png"/>
    <MenuItem key="User" imageSource="../../content/images/Menu/User64x64.png"/>
  </MenuGroup>
  <MenuGroup key="MolecularBiology">
    <MenuItem key="Gene" imageSource="../../content/images/Menu/Gene256x256.png"/>
    <MenuItem key="IntergenicRegion" imageSource="../../content/images/Menu/IntergenicRegion256x256.png"/>
    <MenuItem key="Organism" imageSource="../../content/images/Menu/Organism128x128.png"/>
    <MenuItem key="PredictorMethod" imageSource="../../content/images/Menu/PredictorMethod64x64.png"/>
    <MenuItem key="Role" imageSource="../../content/images/Menu/Role256x256.png"/>
  </MenuGroup>
  <MenuGroup key="Other">
    <MenuItem key="City" imageSource="../../content/images/Menu/City.png"/>
    <MenuItem key="Country" imageSource="../../content/images/Menu/World64x64.png"/>
    <MenuItem key="ImportQueue" imageSource="../../content/images/Menu/Import128x128.png"/>
    <MenuItem key="Publication" imageSource="../../content/images/Menu/Article64x64.png"/>
    <MenuItem key="State" imageSource="../../content/images/Menu/City.png"/>
  </MenuGroup>
</Menu>
```

**Figura 46: Arquivo de configuração dos menus**

Em detalhes, o *MenuProvider* (Figura 47) abre o arquivo XML de configuração (A) e cria um objeto *MenuGroup* para cada nodo XML de mesmo nome localizado (B). Em seguida são percorridos todos os nodos *MenuItem* do grupo no arquivo XML, são verificadas as permissões do usuário para cada item (C), são criados os objetos *MenuItem* e os mesmos são adicionados à coleção de itens do grupo (D). Por fim, é retornada uma lista com todos os grupos carregados com seus devidos itens de menu (E).

<sup>49</sup> *Cookie*: Dado utilizado pelo navegador para armazenar informações da sessão HTTP

<sup>50</sup> XML: *Extensible Markup Language*



```

#region Public Methods

public static List<MenuGroup> LoadMenu(List<AdminGroupPermission> permissions = null)
{
    var groups = new List<MenuGroup>();

    (A) using (var stream = Assembly.GetExecutingAssembly().
        GetManifestResourceStream(MENUCONFIGPATH))
        {
            foreach (XElement serializedGroup in XElement.Load(stream).
                Elements("MenuGroup"))
            {
                (B) var groupKey = serializedGroup.GetAttributeValue("key");
                var group = new MenuGroup(groupKey);

                foreach (var serializedItem in serializedGroup.Elements("MenuItem"))
                {
                    var itemImage = serializedItem.GetAttributeValue("imageSource");
                    var itemKey = serializedItem.GetAttributeValue("key");

                    (C) var permission = GetPermission(permissions, itemKey);

                    var item = new MenuItem(itemKey, itemImage, permission);

                    (D) group.Items.Add(item);
                }

                groups.Add(group);
            }
        }

    (E) return groups;
}

#endregion

```

**Figura 47: Detalhes da classe de serviço MenuProvider**

Cada item de menu possui dois dados básicos em sua estrutura: um identificador (*key*) responsável por identificar qual menu está sendo acessado na área de manutenção e nos cadastros; e o caminho da imagem (*imageSource*) que é a representação visual do item de menu na tela de administração.

O *MenuProvider* é utilizado dentro da operação *Index* do *AdministrationController* (Figura 48) para a construção do menu, e consumido dentro da *View*<sup>51</sup> principal de administração. O funcionamento da operação *Index* é relativamente simples, primeiramente ela verifica se o usuário que está acessando o sistema é um administrador (A), em seguida ela recupera as permissões deste usuário e cria os menus (B).

<sup>51</sup> *View*: Elemento conceitual do padrão MVC



```

public ActionResult Index()
{
    (A) if (!this.SessionManager.CurrentUserIsAdmin())
        return View("Login");

    (B) var permissions = this.SessionManager.GetCurrentUserPermissions();
        var menu = MenuProvider.LoadMenu(permissions);

        return View(menu);
}

```

**Figura 48: Detalhes de implementação do *MenuProvider***

### 6.1.3 Manutenção

A tela de manutenção é compartilhada por todos os itens de menu dá área de administração. Para identificar quais dados deverão ser carregados, é enviada a *key* de cada item de *menu* para a operação *Maintenance* (Figura 49) do *AdministrationController*. Esta operação verifica se o usuário é administrador (A), obtém as permissões do usuário para aquele item e cria o *Gateway* (DAVANZO, 2010) com base na *key* informada (B). Por final, é efetuada uma consulta paginada, retornando a primeira página com 14 registros (C).

```

[AcceptVerbs(HttpVerbs.Get)]
public ActionResult Maintenance(string key)
{
    (A) if (!this.SessionManager.CurrentUserIsAdmin())
        return View("Login");

    (B) var permission = this.SessionManager.GetCurrentUserPermissionFor(key);
        using (var gateway = DataGatewayFactory.GetDataGateway(key))
        {
            (C) var pagedResult = gateway.FindPage(new PageInfo(14, 1));
                return View(new MaintenanceViewModel(pagedResult)
                {
                    Key = key,
                    Page = 1,
                    Permission = permission == null ? PermissionValue.All
                        : (PermissionValue)permission.PermissionLevel
                });
        }
}

```

**Figura 49: Detalhes de implementação da operação *Maintenance***

Para habilitar a *Grid* de manutenção com a função de paginação, foi alterada a classe *DBDataGateway* do *framework LP.Framework* (DAVANZO, 2010), incluindo uma nova operação chamada *FindPage* que recebe como parâmetro um objeto do tipo *PageInfo*, que possui as informações da página atual, e no retorno obtém-se um objeto do tipo *PagedResult*, que contém as informações sobre o resultado paginado.

### 6.1.4 Cadastros

Diferente do que foi projetado inicialmente, os cadastros foram construídos utilizando janelas de dialogo modais e não janelas do navegador. A modificação deste comportamento foi elaborada para reduzir a transição repetitiva entre interfaces, visto que em janelas modais os cadastros abrem sobre a tela de manutenção.

Outro elemento importante no desenvolvimento dos cadastros foi a criação do controle de seleção de relacionamentos entre as entidades, o *picklist*. O mesmo foi criado utilizando funções *javascript* para tratar todos os comportamentos necessários para o seu funcionamento, como a abertura da janela de seleção, a carga dos dados utilizando *AJAX*<sup>52</sup>, e a função de auto-completar. Alguns detalhes de implementação podem ser visualizados nas figuras 50 e 51.

```
Ⓐ function getPickListPageData(pageNumber) {  
  Ⓑ var referenceKey = getReferenceKey(currentPickList);  
  Ⓒ $.post("/Admin/GetPickListPageData",  
    {  
      key: referenceKey,  
      pageNumber: pageNumber,  
      searchColumn: "",  
      searchValue: ""  
    }, function (data) {  
      pickListSelectedId = null;  
      pickListDisplayValue = "";  
      Ⓓ $("#pickListHolder").html(data.Data);  
      loadDisplayColumnIndex();  
      Ⓔ $("#pickListHolder").dialog("open");  
      prepareToUpdateTableLayout("#pickListHolder table");  
    });  
}
```

**Figura 50:** Carga dos dados da janela de seleção do *picklist* (*javascript*)

A *grid* de seleção do *picklist* é carregada através de uma chamada à função *javascript* chamada *getPickListPageData*, que recebe como argumento o número da página a ser carregada (A). Esta função recupera a *key* do cadastro que deverá ser carregado (B), e executa uma função chamada *\$.post* do framework *jQuery*<sup>53</sup> (C) para recuperar a página utilizando *AJAX*. Quando o resultado é obtido, o mesmo é utilizado como conteúdo do elemento *html* *pickListHolder* (D), e a janela de seleção do *picklist* é aberta (E).

<sup>52</sup> *AJAX: Asynchronous Javascript and XML*

<sup>53</sup> <http://jquery.com/>

```

(A) [AcceptVerbs(HttpVerbs.Post)]
public ActionResult GetPickListPageData(string key, int pageNumber, string searchColumn, string searchValue)
{
    if (!this.SessionManager.CurrentUserIsAdmin())
        return Json(null);

    (B) var result = this.GetPageData(key, pageNumber, searchColumn, searchValue);
    var columns = new List<WebGridColumn>(result.Columns.Select(c =>
        new WebGridColumn
        {
            (C) ColumnName = c.Key,
                Header = c.Value,
                Style = "tableColumn",
                Format = (o) => WebGridHelper.CreateDataLink(o.Value, c.Key, 20)
        }));

    (D) var pager = WebGridHelper.CreatePager(pageNumber, result.TotalPages, "getPickListPageData")
        .ToString().Replace("\\", string.Empty);

    WebGridColumn column = new WebGridColumn
    {
        Format = o => WebGridHelper.CreateSelectionLink(key, o.Value),
        Style = "maintenanceActionLink"
    };
    columns.Insert(0, column);

    StringBuilder builder = new StringBuilder();

    (E) var searchBuilder = this.CreateSearchControl(result, searchColumn, searchValue);
    builder.Append(searchBuilder);

    (F) var gridData = this.CreateGridView(result, columns, pager);
    builder.Append(gridData);

    (G) return Json(new { Data = builder.ToString() });
}

```

**Figura 51: Carga dos dados da janela de seleção do *picklist* (*AdministrationController*)**

O método *GetPickListPageData* do *controller AdministrationController* (A) é executado em resposta a função *\$.post* demonstrado na Figura 50. O mesmo tem como funções principais: recuperar os dados paginados de acordo com os argumentos do método (B), criar as colunas que serão apresentadas no *grid* (C), criar o *html* do paginador da *grid* (D), criar o *html* do controle utilizado para filtro (E) e criar o *html* da *grid* de resultado (F). Todos os controles criados são combinados, e é retornado um *string*<sup>54</sup> com a *grid* de resultados (G).

## 6.2 AUTENTICAÇÃO E AUTORIZAÇÃO DE USUÁRIOS

Para efetuar a manutenção de usuários do sistema e grupos de administradores, foram utilizados os recursos da área de administração. Contudo, para gerenciar os acessos às áreas restritas do sistema, foram criados recursos novos como o *SessionController* e o *ControllerSessionManager*.

<sup>54</sup> *String*: Cadeia de caracteres

O *SessionController* é uma classe abstrata simples, que através do mecanismo de herança, herda de *Controller*<sup>55</sup> e provê um objeto do tipo *ControllerSessionManager*.

A classe *ControllerSessionManager* disponibiliza uma série de funções para identificar o tipo de usuário que está executando a aplicação, se o usuário é administrador, e as permissões que o administrador tem a nível de cadastro (incluir, excluir, editar e visualizar).

Os *controllers* que herdam de *SessionController* são o *AdministrationController*, para garantir que os usuários são administradores, o *DefaultController* para assegurar que os usuários estão ativos no sistema, e o *AccountController*.

Com o intuito de assegurar a veracidade do *e-mail* dos usuários cadastrados na página principal do IntergenicDB, é enviado uma solicitação de ativação da conta para o mesmo. Para enviar esta solicitação, foi criado um serviço responsável por enviar *e-mails* chamado *EmailService*. O mesmo necessita uma conta de um servidor SMTP<sup>56</sup> que deve ser configurada no arquivo *web.config* (Figura 52) da aplicação.

```
<appSettings>
  <add key="ClientValidationEnabled" value="true"/>
  <add key="UnobtrusiveJavaScriptEnabled" value="true"/>
  <add key="ImportDirectoryPath" value="..\Temp"/>
  <add key="SMTPServerAddress" value="smtp.gmail.com"/>
  <add key="SMTPPassword" value="password"/>
  <add key="SMTPUser" value="douglas.picolotto@gmail.com"/>
  <add key="SMTPSSL" value="false"/>
</appSettings>
```

**Figura 52: Configuração SMTP no web.config**

As informações necessárias para a configuração do *e-mail* são: o endereço SMTP do servidor de *e-mails*, o usuário e a senha utilizada para efetuar a autenticação neste servidor e um parâmetro informando se a opção de segurança SSL<sup>57</sup> será utilizada (A).

### 6.3 UPLOAD DE ARQUIVOS

O classe de serviço responsável pelo processo de importação chama-se *IntergenicDatabaseService*. Através da operação *ImportData* da mesma, os arquivos eram processados, o resultado era armazenado em tabelas temporárias, e em seguida era utilizada a

<sup>55</sup> *Controller*: Parte do framework ASP .Net MVC

<sup>56</sup> SMTP: *Simple Mail Transfer Protocol*

<sup>57</sup> SSL: *Secure Sockets Layer*

*StoredProcedure ImportData* para transferir os dados das tabelas temporárias para as tabelas finais. Todo o processo ocorria em uma única etapa. Para efetuar a separação do mesmo foi removida a chamada á *StoredProcedure* da operação *ImportData*. A nova versão da primeira etapa de importação possui três responsabilidades: criar uma nova fila de importação, salvar o arquivo importado no sistema de arquivos do servidor e carregar os dados em tabelas temporárias.

O restante do processo de importação de arquivos é efetuado utilizando os recursos da área de administração, porém o grande diferencial é que os dados não são salvos diretamente no banco de dados, é chamada outra operação do serviço *IntergenicDatabaseService* chamada *ArrangeData* (Figura 53), que tem como principal função armazenar os dados nas tabelas finais e remover a fila de importação já processada.

```
(A) public int ArrangeData(uint id)
{
    using (IDbCommand cmd = DataGateway.CreateCommand())
    {
(B) cmd.CommandText = "ImportData";
        cmd.CommandType = CommandType.StoredProcedure;

        var parameter = cmd.CreateParameter();
        parameter.ParameterName = "_importQueueId";
        parameter.Value = id;
        cmd.Parameters.Add(parameter);

        if (Connection != null)
            cmd.Connection = Connection;

        if (cmd.Connection.State != ConnectionState.Open)
            cmd.Connection.Open();

        int result = 0;

        try
        {
            result = cmd.ExecuteNonQuery();
(C) using (var gateway = DataGatewayFactory.GetDataGateway<ImportQueue>())
                gateway.Delete(new ImportQueue { ImportQueueId = id });
        }
        catch
        {
            throw;
        }

        return result;
    }
}
```

**Figura 53: Detalhes de implementação da operação *ArrangeData***

O método *ArrangeData* recebe como argumento o identificador do objeto *ImportQueue* (A). Para efetuar o processamento do *ImportQueue* é executada a *StoredProcedure ImportData* (B), que irá ler os dados temporários armazenados nas tabelas

*ImportQueue* e *ImportQueueEntry* e salvar nas tabelas do IntergenicDB. Caso a execução ocorra com sucesso, é utilizado um objeto *Gateway* do *LP.Framework* para excluir o *ImportQueue* processado.

#### 6.4 CARGA DOS DADOS

Para a carga dos dados relacionados a regiões intergênicas, foram utilizados os arquivos providos pela profa. Dra. Scheila de Ávila e Silva já no formato dos arquivos de importação no IntergenicDB. Desta forma, foi utilizada a própria ferramenta de *upload* de arquivos para efetuar a carga.

Em relação aos dados de predição, a carga foi executada conforme descrita na etapa de projeto deste trabalho. Para isto, foi criado um programa específico que efetua uma consulta no *Prompredict*<sup>58</sup> (Figura 54) para cada região intergênica existente no banco de dados do IntergenicDB, armazenando o resultado em um arquivo temporário, utilizando como nome do mesmo o identificador da região intergênica no banco de dados.

#File	Format#ID	sequence identifier	Length of the squnce	%GC of sequence#>pstart..pend	
length of the promoter	promoter seq	lsp	lspe	Dmax pos	Dmax
Dave#	#####	#####	#####	#####	#####
ID	Header	+			
572	36.89				
ID	Header	-			
572	36.89>58..82	25	aagttataaaattcacatttgccat	64	-13.52 66 2.20 1.90

**Figura 54: Formato do arquivo do *Prompredict***

Depois de finalizada a etapa de consultas ao *Prompredict*, estes arquivos são processados para obter a sequência de nucleotídeos identificada na predição e a posição da mesma na região intergênica. Por último, este resultado é armazenado no banco de dados na tabela *IntergenicSlice*, e vinculado ao registro do método de predição do *Prompredict* no banco de dados.

<sup>58</sup> <http://nucleix.mbu.iisc.ernet.in/prompredict/prompredict.html>



## 6.5 ÁREA DE POSTAGEM

Para a criação da área de postagem, foi incluída uma nova aba na *Home* do IntergenicDB chamada *News* (Figura 55), utilizando o método *ActionLink* de um facilitador chamado *Html* disponibilizado pelo ASP.Net MVC (A).

```
<ul>
  <li>@Html.ActionLink("Home", "Index", "Default")</li>
  <li>@Html.ActionLink("Search", "Search", "Default")</li>
  (A) <li>@Html.ActionLink("News", "News", "Default")</li>
  @if (Request.IsAuthenticated)
  {
    <li>@Html.ActionLink("Upload", "Upload", "Default")</li>
  }
  <li>@Html.ActionLink("Help", "Help", "Default")</li>
</ul>
```

**Figura 55: Detalhes de implementação da aba *News***

Em seguida, foi desenvolvida a *View* que é apresentada quando a aba *News* é selecionada. Esta *View* é composta por uma lista de *links*, um paginador e um mecanismo de busca. Um detalhe importante sobre estes elementos, é que o paginador foi criado como um componente e é utilizado em diversos locais da aplicação, como na *grid* de Manutenção, no *Picklist*, na aba *Search* e na aba *News* (Figura 56). O método *CreatePager* recebe como parâmetros a página corrente, o total de páginas e o nome da função *javascript* que será utilizada para efetuar a paginação.

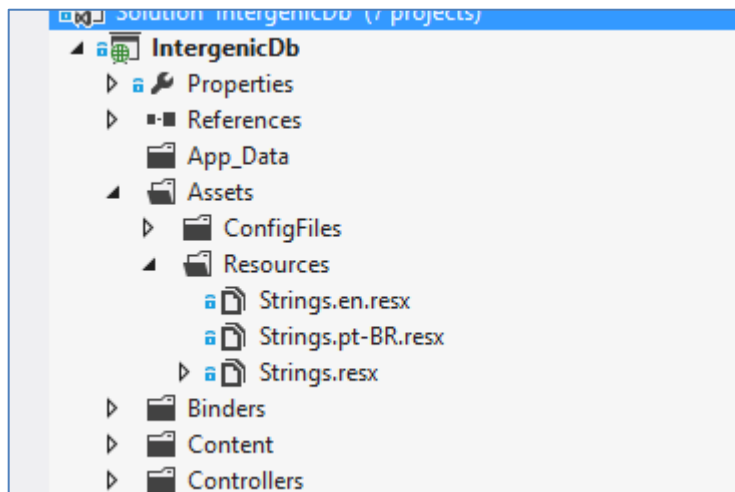
```
<div id="newsFooter">
  @IntergenicDb.Helpers.WebGridHelper.CreatePager(Model.Result.CurrentPage,
  Model.Result.TotalPages,
  "searchNews")
</div>
```

**Figura 56: Detalhes da utilização do paginador na aba *News***

## 6.6 INTERNACIONALIZAÇÃO

Para adicionar o suporte a internacionalização ao IntergenicDB foram utilizados recursos específicos do *.Net Framework*, chamados *Resources* (Figura 57). Basicamente, *resources* são arquivos XML que armazenam conteúdos diversos, e são gerenciados por um serviço chamado *ResourceManager*.

O nome dos arquivos de *Resources* possui um formato específico formado por: {NomeDoArquivo}.{Língua}.{resx}. Os arquivos são selecionados de acordo com a língua utilizada na *Thread*<sup>59</sup> em execução.



**Figura 57: Arquivos de *Resources***

Para recuperar os valores traduzidos, foi desenvolvido um serviço específico que acessa o *ResourceManager* do arquivo de *Resources*, e utiliza a operação *GetString* (Figura 58) passando como parâmetro o valor da chave armazenada no arquivo (*value*).

```
public static class GlobalizationHelper
{
    public static string GetLocableString(string value)
    {
        var result = Strings.ResourceManager.GetString(value);
        return result ?? value;
    }
}
```

**Figura 58: Detalhes do serviço *GlobalizationHelper***

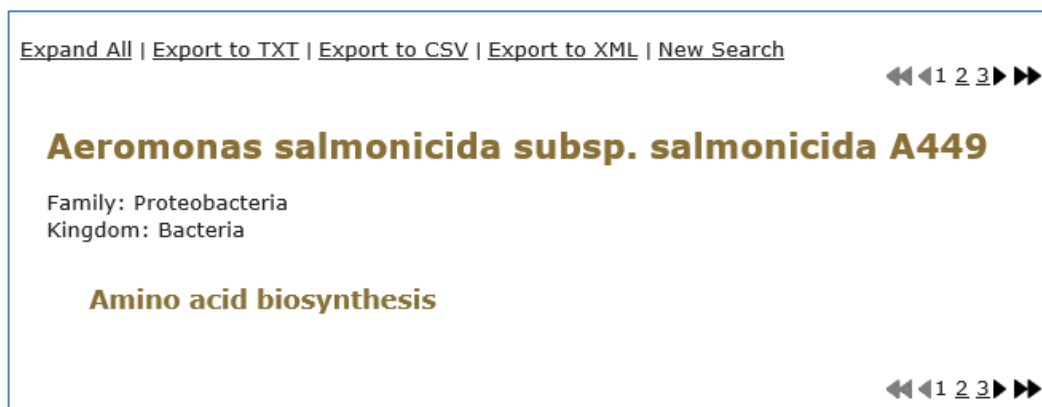
## 6.7 CONSULTAS E TESTES

A primeira modificação efetuada na aba *Search* foi a inclusão de um mecanismo de paginação para a apresentação dos resultados (Figura 59). A paginação é necessária devido a problemas de desempenho na geração dos elementos visuais pelo navegador, quando a quantidade de dados retornados for relativamente grande. É importante salientar que este

<sup>59</sup> *Thread*: Em informática, possibilita dividir um processo permitindo a execução de diversas tarefas concorrentemente.



paginador existe somente para os resultados apresentados no navegador, o *download* de arquivos não possui este mecanismo.



**Figura 59: Tela de resultados com paginação**

Em seguida foi incluída uma nova opção de consulta por método de predição (Figura 60). Foi utilizado o método *TextBoxFor* do facilitador *Html* para criar os campos de pesquisa por método de predição (A e B). Este método cria uma caixa de texto em *html* utilizando o nome da propriedade como identificador e nome do campo.

```
<p><a href="#predictorMethod" class="toggleable">Predictor Method</a></p>
<div id="predictorMethod" class="search-box">
  Name:
  (A) @Html.TextBoxFor(m => m.PredictorMethodName)
  <br />
  Hit Percentage:
  (B) @Html.TextBoxFor(m => m.PredictorMethodHitPercentage, new { @class = "margin-right" })
</div>
```

**Figura 60: Detalhes de implementação da aba *Search***

Os testes de desempenho e de resultado serão apresentados no capítulo sete deste trabalho.

## 6.8 DIFICULDADES ENCONTRADAS

A principal dificuldade encontrada foi a falta de conhecimento sobre a tecnologia utilizada. Este problema tornou o desenvolvimento muito oneroso e cansativo, principalmente durante os primeiros meses de desenvolvimento.

Outro grande problema foi em relação a utilização de componentes de terceiros. Como referência, cabe citar o controle *WebGrid* do próprio *ASP.Net MVC*, que impossibilitou a utilização do paginador embutido no próprio controle, tornando necessária a criação de um paginador customizado.

Por último, mas não menos problemático, existe o problema de *Cross-Browser*<sup>60</sup>, onde funções *javascript*<sup>61</sup> e estilos em *css*<sup>62</sup> têm comportamentos diferentes em navegadores diferentes. Por este motivo, os navegadores homologados para o IntergenicDB são o *Microsoft Internet Explorer*<sup>63</sup> versão 10, o *Google Chrome*<sup>64</sup> e o *Mozilla Firefox*<sup>65</sup>.

## 6.9 CONSIDERAÇÕES FINAIS

Salvo algumas alterações efetuadas para dar suporte a paginação, pesquisas nos cadastros e as novas funcionalidades, foi possível utilizar praticamente toda a infraestrutura desenvolvida na primeira versão do IntergenicDB.

Em relação ao projeto desenvolvido no capítulo anterior, parte da modelagem desenvolvida na etapa de projeto precisou ser revisitada, devido a falta de conhecimento em relação a tecnologia utilizada e imprevistos que surgiram durante o desenvolvimento já citados na seção anterior. Como parte da entrega do sistema, no ANEXO F deste trabalho foi descrito o processo de instalação do IntergenicDB no servidor.

No próximo capítulo são apresentados testes de desempenho no mecanismo de consultas do IntergenicDB, que é a principal função do portal.

---

<sup>60</sup> *Cross-Browser*: Suporte a múltiplos navegadores

<sup>61</sup> <http://www.w3schools.com/js/default.asp>

<sup>62</sup> *Css*: *Cascading Style Sheets*

<sup>63</sup> <http://www.microsoft.com/pt-br/download/ie.aspx?q=internet+explorer>

<sup>64</sup> <http://www.google.com/intl/pt-BR/chrome/browser/>

<sup>65</sup> <http://br.mozdev.org/>

## 7 ESTUDO DE CASO

Neste capítulo serão demonstradas algumas consultas sendo processadas, bem como um resultado tabular sobre o processamento das mesmas. Os testes de desempenho serão efetuados apenas na exportação dos resultados obtidos, pois os resultados apresentados no navegador são paginados, impossibilitando que resultados reais sejam gerados.

### 7.1 AMBIENTE DE TESTES

Os testes serão efetuados em um computador com processador Intel *Core I5*<sup>66</sup>, 8MB de memória RAM e 500GB de HD<sup>67</sup>. O servidor *Web* utilizado é o *Microsoft Internet Information Services 8*<sup>68</sup>. A base de dados utilizada para os testes é composta por informações de 7 organismos procariotos, somando um total de 24150 genes e 10869 regiões intergênicas. Para todos os resultados serão apresentadas as telas, e as informações “tempo de processamento” e “tamanho de arquivo” geradas pela exportação em arquivo texto (TXT).

### 7.2 CONSULTAS SIMPLES

As consultas simples foram efetuadas utilizando apenas uma opção de filtro por vez.

#### 7.2.1 Organismo

No IntergenicDB é possível efetuar pesquisas de organismos utilizando como critério de busca nome, reino e família. O resultado obtido da pesquisa do organismo de nome “*Escherichia coli* APEC O1” foi um arquivo de 985KB<sup>69</sup> em apenas 0.3441264 segundos. Utilizando o valor “Bacteria” para o critério de pesquisa reino, o resultado obtido foi um arquivo de 4147 KB em 1.4474572 segundos. E por fim, utilizando o valor “Proteobacteria” como valor do critério família, o resultado foi um arquivo de 3643 KB em apenas 1.2124185 segundos. A tela resultante desta consulta pode ser visualizada na figura 61.

---

<sup>66</sup> <http://www.intel.com.br/content/www/br/pt/processors/core/core-i5-processor.html?cid=sem45p16962>

<sup>67</sup> HD: *Hard Disk*

<sup>68</sup> <http://www.iis.net/>

<sup>69</sup> Kb: *Kilobyte*

## Escherichia coli APEC O1

Família: Proteobacteria  
Reino: Bacteria

### Amino acid biosynthesis

- conserved hypothetical protein

GC: 48,83 %

Comprimento: 861

Direção: F

Posição: 2898129

Sequência:

```
AGAAACCTGAAACACAAAACGGCAGCCTTGAGCTGCCGTTTTTTTATTCTGTTAGTTGTG
AAACTGAAGCGATTTAGTCACTATCGATCTCATCAAATATGGCTCGCTTTGAGATATTCC
TCAAGTAAAAAACACCTCTTCCTGCGATTTCTCACAAAAAAGATTCTGACAAAAAGT
GACAAAATTATGAGATTTCCATCACACATTTTGACATCAGGAACGTTATGCTGAATTCAC
CAAGACGGGAAGACAAGAGGTAAAATTTATGACAATGAACATTACCAGCAAACAAATGGA
AATTACTCCGGCCATCCGCCAACATGTCGCAGACCGTCTCGCCAAACTGGAAAAATGGCA
AACACATCTGATTAATCCACATATCATTCTGTCCAAAGAGCCACAAGGGTTTGTGTCTGA
CGCCACAATCAATACACCTAACGGCGTTCTGGTTGCCAGTGGTAAACATGAAGATATGTA
CACCGCAATTAACGAATTGATCAACAAGCTGGAACGGCAGCTCAATAAACTGCAGCACAA
AGGCGAAGCACGTCGTGCCGCAACATCGGTGAAAGACGCCAACTTCGTGGAAGAAGTTGA
AGAAGAGTAGTCCTTTATATTGAGTGTATCGCCAACGGCCTTCGGGGCGCGTTTTTTGTT
GACAGCGTGAAAACAGTACGGGTACTGTAATAAGTCACTTAAGGAAACAAACATGAAAC
ACACACCGTTTTTCTTCGCATTCTTTTTACCTTCCCCTGAATGGGAGGCGTTTCGTCTGT
GTGAAACAGAATGCGAAGACGAACAATAAGGCCTCCCAAATCGGGGGGCCTTTTTTTATTG
ATAACAAAAAGGCAACACTAT
```

Métodos de Predição

Figura 61: Seção do resultado de pesquisa por nome do organismo

### 7.2.2 Gene

A pesquisa por gene do IntergenicDB possibilita consultas utilizando nome, função principal (*mainrole*), símbolo e %GC. Contudo, para os testes efetuados foram utilizados apenas os dois primeiros.

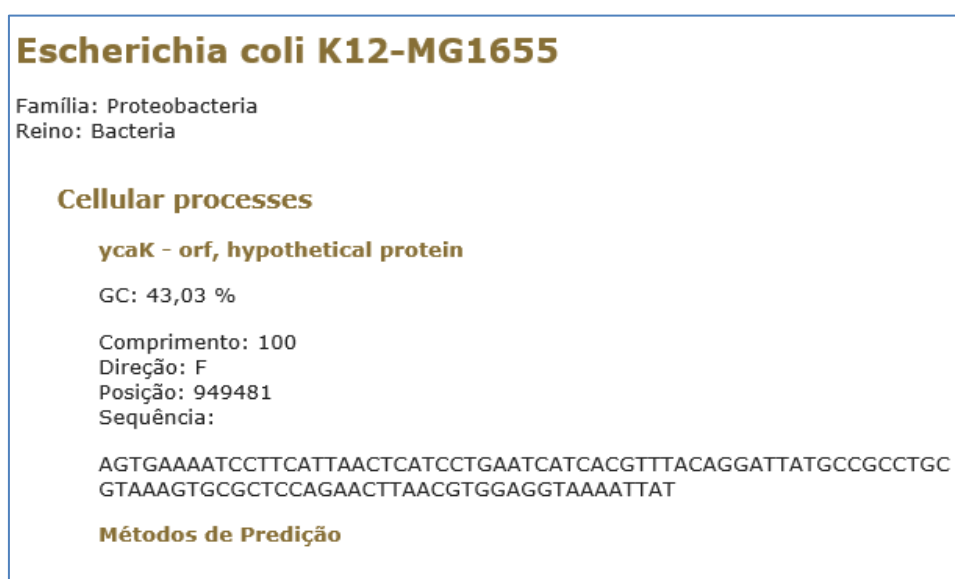
O tempo de processamento de uma consulta utilizando o valor “*transcriptional regulator AsnC*” para o critério nome, foi de apenas 0.0820257 segundos e como resultado foi obtido um arquivo de 2 KB (Figura 62). A pesquisa por função principal do gene foi efetuada utilizando o valor “*Transcription*”, e como resultado foi obtido um arquivo de 50 KB em 0.0870299 segundos.



**Figura 62: Resultado de pesquisa por nome do gene**

### 7.2.3 Região Intergênica

A pesquisa por região intergênica foi efetuada utilizando os critérios comprimento e fita. Utilizando o valor “100” para o primeiro, o resultado foi um arquivo de 14KB com tempo de processamento muito pequeno, aproximadamente 0.0330217 segundos (Figura 63). O segundo resultou em um arquivo de 1775KB em um tempo de 0.7202277 segundos, quando utilizado o valor “Reverse”.



**Figura 63: Seção do resultado de pesquisa por comprimento da Região Intergênica**

## 7.2.4 Método de Predição

A pesquisa por método de predição é uma das novas funcionalidades do IntergenicDB. O método é mostrado no resultado logo abaixo da região intergênica localizada na pesquisa. As informações contidas no resultado do método são: o percentual de acerto do método, a posição da sequência identificada e a própria sequência promotora (Figura 64).

Para o teste executado desta nova opção de consulta, foi utilizado o nome do método de predição com o valor “DNA Stability”, o resultado obtido foi um arquivo de 1067KB com um tempo de processamento de 0.2390761 segundos.

**Aeromonas salmonicida subsp. salmonicida A449**

Família: Proteobacteria  
Reino: Bacteria

**Amino acid biosynthesis**

**ilvH - acetolactate synthase isozyme III, small subunit**

GC: 62,62 %

Comprimento: 404  
Direção: R  
Posição: 3683990  
Sequência:

GATTGTTTCGGCAGGTGGATTGGTTATCCCAAAGGGGCTCTTCGGAGCCCTTTTTACGGG  
CGCAAGCTGAGCCAGGGGGAAGTGCGAGTCTGTCATTTTCTTTCAAATGATCAGTAAACT  
ATCCACAAAAGCTGTGGAAAACCTGTAGATTAGCGGTGTCTAGTGTAGTGAGCTGA  
TCTTAAAGGGGTATTTGGATCTGATCTCTGCTGGGGTGCAATTGTGCAATCCCCTGTGGT  
AGGTGTGAGTGTGCTTATCCTGCCAGCCTTGCCAGACAAGCCATGGAGAGAGGCATAG  
TGATCCGGTCAGGATCCTGATTTGATCTTTCGGGATCAACATGTCGCCCGCCCACTATAAA  
AAAACCCGCCTCTCGGCGGGTTTTTCACAGGGTGGCTGAAGGGCG

**Métodos de Predição**

Método de Predição: DNA Stability  
Percentual de Acerto: 56 %

Posição: 108  
Sequência:

TGATCAGTAAACTATCCACAAAAGCTGTGGAAAACCTGTAGATTAGCGG  
TGTGTCTAGTGTAGTGAGCTGATCTTAAAGGGG

**Figura 64: Exemplo de resultado com método de predição**

## 7.3 CONSULTAS COMBINADAS

As consultas combinadas foram efetuadas utilizando dois filtros por pesquisa.

### 7.3.1 Organismo x Gene

Na pesquisa combinada entre organismo e gene foi utilizado como critérios o nome do organismo, com o valor “*Escherichia coli W3110*”, e a função do gene, com o valor

“*Signal transduction*” (Figura 65). Como resultado foi gerado um arquivo de 13KB em aproximadamente 0.0820230 segundos.

### Escherichia coli W3110

Família: Proteobacteria  
Reino: Bacteria

#### Signal transduction

**arcA - DNA-binding response regulator in two-component regulatory system with ArcB or CpxA**

GC: 51,04 %  
Comprimento: 58  
Direção: R  
Posição: 4644270  
Sequência:  
AATCGGCTTTACCACCGTCAAAAAACGGCGCTTTTTAGCGCCGTTTTATTTTTCAAC

**Métodos de Predição**

**Figura 65: Seção do resultado da pesquisa organismo x gene**

### 7.3.2 Gene x Região Intergênica

A pesquisa combinada entre gene e região intergênica foi executada utilizando-se o valor “*two-component system sensor histidine kinase*” para o nome do gene e o valor “*Forward*” para o tipo da fita da região intergênica (Figura 66). Como resultado, foi obtido um arquivo de 4KB em 0.0480179 segundos.

### Aeromonas salmonicida subsp. salmonicida A449

Família: Proteobacteria  
Reino: Bacteria

#### Hypothetical proteins

**- two-component system sensor histidine kinase**

GC: 59,45 %  
Comprimento: 323  
Direção: F  
Posição: 272115  
Sequência:  
CTCACCGATCCTTGGGTTGGGCTCATCTGAAATGTATACACGGCTCACGGTTATGGGATT  
AACTATGCATTAACATACATTTGTCTGTTGTGATCAGGCTCTTGTGGCAAGAAAGGAGG  
TTTGCATGAGTGTGGAAGTAAATGGGCACTGTTGACCGCATCCAGCGCTTGGCAATGA  
TCCTGATTATCTGTGTGGCATCCGTCGGTCTGGCGTGAACCCCGTCCCTCATTGCGGTA  
TGAACCGGCCTCCTGGCCGTTTTGCTTTTGGGGGAGTGCAGATATGCTGACCCACCCAT  
GATGAATCGAGACAGCACAAGAT

**Métodos de Predição**

**Figura 66: Seção do resultado da pesquisa gene x região intergênica**

### 7.3.3 Gene x Método de Predição

Para a pesquisa combinada entre gene e método de predição, foi utilizado o valor “*two-component regulatory system*” para o nome do primeiro e o valor “*DNA Stability*” para o nome do segundo (Figura 67). O resultado foi um arquivo de 5KB gerado em apenas 0.0406501 segundos.

**Escherichia coli APEC 01**

Família: Proteobacteria  
Reino: Bacteria

**Central intermediary metabolism**

***glnG - fused DNA-binding response regulator in two-component regulatory system with GlnL: response regulator/sigma54 interaction protein***

GC: 56,38 %

Comprimento: 358  
Direção: R  
Posição: 4371950  
Sequência:

GATTCACGGCTTGTGTGTAAGATTGATTATTGAGCGCAAATTGCTGGTATTTGCGCTT  
TACTGTTCCGATAAGTTCAGTATGATCTTGCCCGGAAAACGGGGAGAGTCATTATGCTGG  
AATCAATAATTAATCTGGTATCGAGTGGCGCAGTTGACAGCCACACACCGCAAAGTCTG  
TTGCTGCCGTGCTGTGTGCCGCGATGATAGGGCTGTTTAGCTGAAGCTCGTTGTTACGC  
CGGATGCGGAGAAAAGTCGTAGGCCTGATAAGACGCAACCAGCGTCGCATCAGGCAATCTT  
GTTCTCAGCACATTGATGGCGCTTTCGTTTTCTACTTTGTAACGAAGCGCCATTACAA

**Métodos de Predição**

Método de Predição: DNA Stability  
Percentual de Acerto: 56 %

Posição:92  
Sequência:  
CCG

**Figura 67: Seção do resultado da pesquisa gene x método de predição**

## 7.4 CONSIDERAÇÕES FINAIS

Mesmo com uma base de dados relativamente pequena, foi possível efetuar as consultas propostas e validar os resultados obtidos, tomando por base os valores utilizados como critérios de pesquisa. Os dados gerados pela exportação de arquivos TXT podem ser visualizados de forma resumida na Tabela 2.



**Tabela 2: Resultados dos testes de exportação das consultas em TXT**

	<b>Crítérios de Pesquisa</b>	<b>Dados utilizados</b>	<b>Tamanho do arquivo TXT</b>	<b>Tempo (em segundos)</b>
<b>Organismo</b>	Nome	Escherichia coli APEC O1	985KB	0.3441264
	Reino	Bacteria	4147KB	1.4474572
	Família	Proteobacteria	3643KB	1.2124185
<b>Gene</b>	Nome	transcriptional regulator AsnC	2KB	0.0820257
	Função Principal	Transcription	50KB	0.0870299
<b>Região Intergênica</b>	Comprimento	100	14KB	0.0330217
	Fita	Reversa	1775KB	0.7202277
<b>Método de Predição</b>	Nome	DNA Stability	1067KB	0.2390761
<b>Organismo x Gene</b>	Nome do Organismo + Função do Gene	Escherichia coli W3110 + Signal transduction	13KB	0.0820230
<b>Gene x Região Intergênica</b>	Nome do Gene + Fita	two-component system sensor histidine kinase + Para a frente	4KB	0.0480179
<b>Gene x Método de Predição</b>	Nome do Gene + Nome do Método	two-component regulatory system + DNA Stability	5KB	0.0406501

## 8 CONCLUSÃO

Como o objeto de estudo deste trabalho já estava em desenvolvimento, foi de suma importância o estudo efetuado sobre biologia molecular, que proporcionou a base de conhecimento sobre a ciência abordada pelo projeto. Para fundamentar a relação entre informática e biologia molecular foi efetuado um estudo sobre bioinformática, que auxiliou no entendimento das funcionalidades de diversos portais de propósito similar. Depois de efetuados estes estudos, o IntergenicDB passou por uma etapa de análises e testes para conhecer a versão existente. Com o conhecimento obtido, foi possível satisfazer todos os objetivos específicos propostos neste trabalho.

Foram elaborados os requisitos vistos como necessários para a utilização do portal em conjunto com a profa. Dra Scheila de Ávila e Silva. Alguns deles alteravam comportamentos existentes visando aprimorar o processo atual, como é o caso do processo de *upload* de arquivos, e outros foram elaborados desde a concepção, como a área de administração do portal. As ferramentas para a prototipação das interfaces, desenvolvimento da aplicação e atualização do banco de dados foram selecionadas de acordo com a necessidade e as tecnologias utilizadas. Em seguida, foi elaborado um projeto que contemplou todos os requisitos identificados, especificando o visual das futuras funcionalidades bem como o seu comportamento. Todos os requisitos previstos foram desenvolvidos, alguns sofrendo modificações durante o processo, devido a alguns fatores como a falta de conhecimento de parte do *framework* utilizado, e outros foram criados tal e qual foram projetados. Imprevistos foram encontrados durante o desenvolvimento, o que tornou o mesmo mais oneroso do que o esperado, contudo todos foram contornados. Ao final do processo de desenvolvimento, foram efetuados testes no sistema para validar as novas implementações, em especial na funcionalidade de consulta.

O IntergenicDB tem capacidade de tornar-se um grande portal pois, em relação aos demais portais estudados, possui um tratamento único sobre as informações relacionadas a regiões intergênicas e promotores. Além disto, as novas funcionalidades proporcionaram ao IntergenicDB uma série de benefícios como: melhor controle sobre informações novas e existentes; melhor gerenciamento e administração do portal, inclusive com possibilidade de governança sobre os administradores; maior facilidade de utilização; e maior integridade dos dados. Por estes motivos, conclui-se que o mesmo já está apto a ser utilizado pela comunidade científica em estudos sobre promotores.

## 8.1 TRABALHOS FUTUROS

Como sugestão, futuramente poderá ser efetuada uma nova carga na base de dados utilizando os arquivos de outros portal já existente, como o *Genbank* ou o *ENA* para ampliar o conjunto de dados do IntergenicDB. Outra atividade importante a ser desenvolvida, é a compatibilização do portal com outros navegadores além do *Internet Explorer* 10. Por último, testes mais rigorosos poderão ser efetuados, especialmente em relação a desempenho, quando o portal estiver em produção, com uma base de dados mais desenvolvida.

## 9 REFERÊNCIAS

ALBERTS, B.; BRAY, D.; JOHNSON, A.; LEWIS, J.; RAFF, M.; ROBERTS, K.; WALTER, P. **Fundamentos da Biologia Molecular: Uma introdução à Biologia Molecular da Celula**. ArtMed, 1999.

ALBERTS, B.; JOHNSON A.; LEWIS J.; RAFF M.; ROBERTS K.; WALTER P.. **Biologia Molecular da Célula**. 5 ed. Artmed, 2010.

DAVANZO, V. **Desenvolvimento de Consultas para um Banco de Dados de Sequências Intergênicas**. Centro de Computação e Tecnologia da Informação, Universidade de Caxias do Sul: Caxias do Sul: 2010. Bacharelado em Ciências da Computação – Trabalho de Conclusão de Curso.

FOWLER, M.; RICE, D.; FOEMMEL, M.; HIEATT, E.; MEE, R.; STAFFORD, R. **Patterns of Enterprise Application Architecture**. Addison Wesley, 2002.

FREEMAN, A.; SANDERSON, S. **Pro ASP.NET MVC 3 Framework**. 3. Ed. Apress, 2011.

GAMMA, E.; HELM R.; JOHNSON, R; VLISSIDES, J. **Design Patterns – Elements of Reusable Object-Oriented Software**. 1 ed. Addison-Wesley Professional, 1994.

GRIFFITHS, A.J.F.; MILLER, J.H.; SUZUKI, D.T.; LEWONTIN, R.C.; GELBART, W.M. **Introdução à Genética**. 7 ed. Guanabara Koogan, 2002. 794p.

JUNQUEIRA L. C.; CARNEIRO J. **Biologia Celular e Molecular**. Guanabara Koogan, 7ª Ed. 2000.

LESK A. M. **Introdução à BioInformática**. 2 ed. Artmed, 2008.

MOLIN, A. F. **Prototipagem de um Banco de Dados de Promotores como Base para um Portal de Serviços**. Centro de Computação e Tecnologia da Informação, Universidade de Caxias do Sul: Caxias do Sul: 2009. Bacharelado em Ciência da Computação – Trabalho de Conclusão de Curso.

PROSDOCIMI, F.; CERQUEIRA, G. C.; BINNECK, E.; SILVA, A. F.; REIS, A. N.; JUNQUEIRA, A. C. M.; SANTOS, A. C. F.; NHANI JR, A.; WUST, C. I.; FOLGUERAS-FLATSCHART, A.V.; SAMPAIO, V. S.; JARDIM, S.; PEREIRA, R. M.; LIMA, R. P.; FERREIRA, R. G. M.; CAMARGO, L. P.; PETRETSKI, J. H.; KESSEDJIAN, J. L.; CAMARGO Fº, F. **Bioinformática: Manual do Usuário**. [Online] Revista Biotecnologia Ciência & Desenvolvimento, 2003. [Citado em: 5 de abril de 2010]  
<http://www.biotecnologia.com.br/revista/bio29/bioinf.pdf>

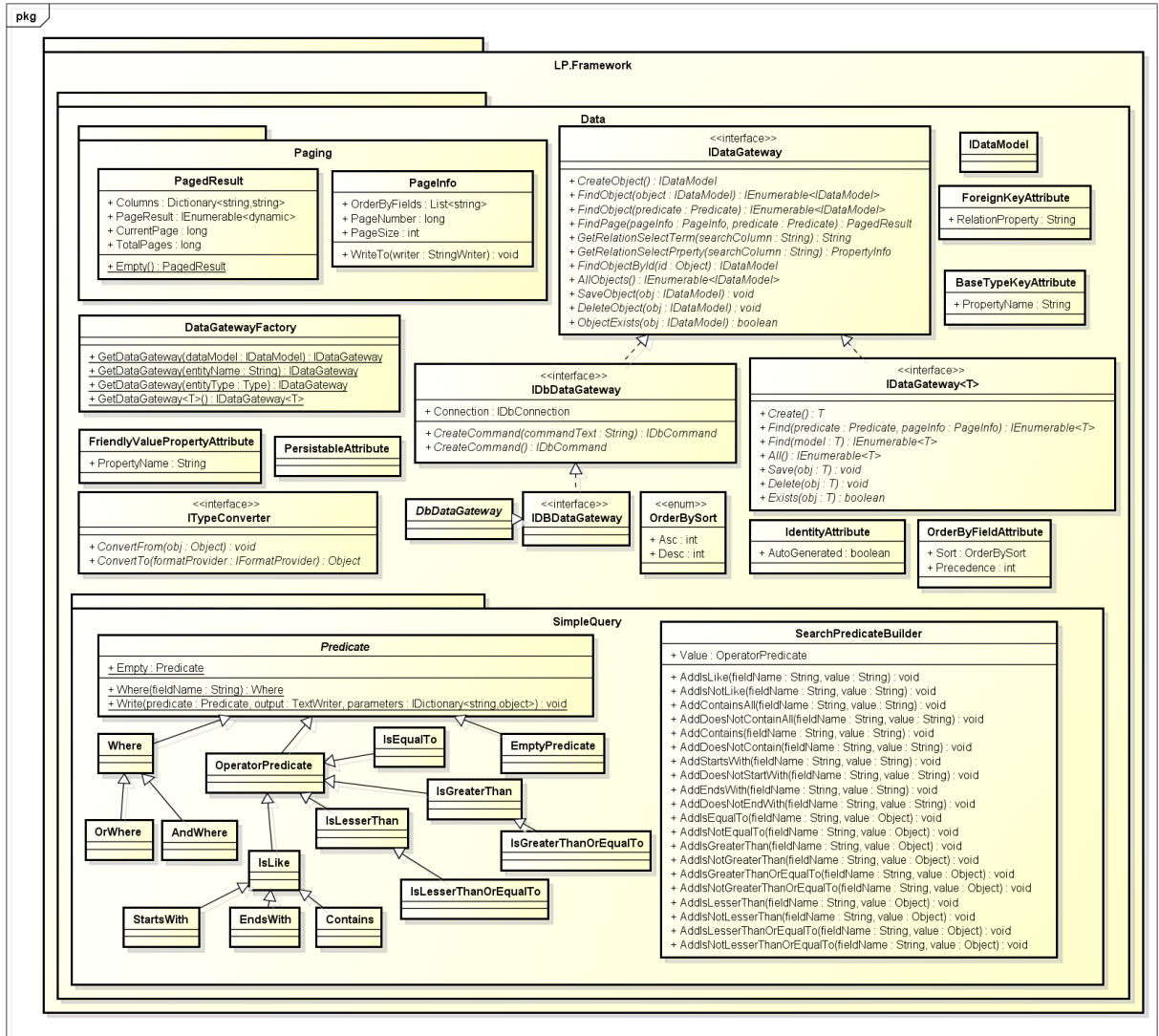
SNUSTAD, P. D; SIMMONS, M. J. **Principles of Genetics**. 3 ed. John Wiley & Sons, Inc., 2003.

VALIATI, J. F. **Redes Neurais Aplicadas ao Reconhecimento de Regiões Promotoras na Família *Mycoplasmataceae***. Instituto de Informática, Universidade Federal do Rio Grande do Sul: Porto Alegre: 2006. Programa de Pós-Graduação em Computação – Tese de Doutorado.  
<http://www.lume.ufrgs.br/bitstream/handle/10183/7590/000549152.pdf?sequence=1>

W3C. **World Wide Web Consortium - Header Field Definitions**. Disponível em: <  
<http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html>>. Acesso em: jun. 2012

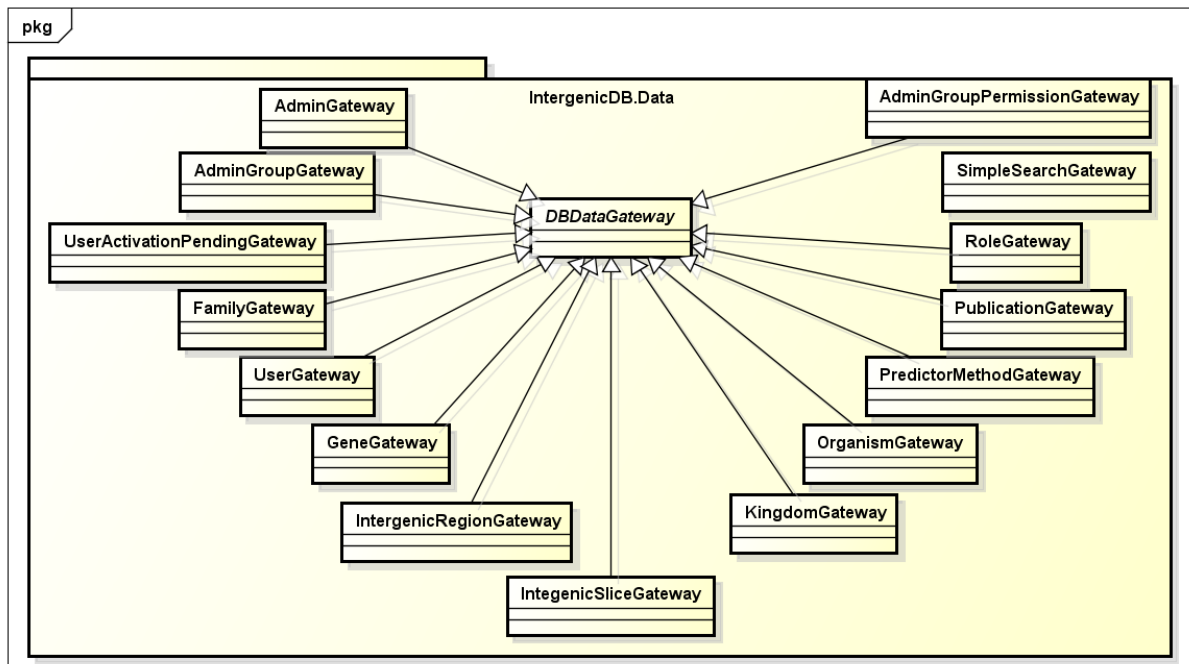
ZAHA, A. **Biologia Molecular Básica**. Mercado Aberto, 1996.

# ANEXO A – Diagrama de classes do projeto LPFramework





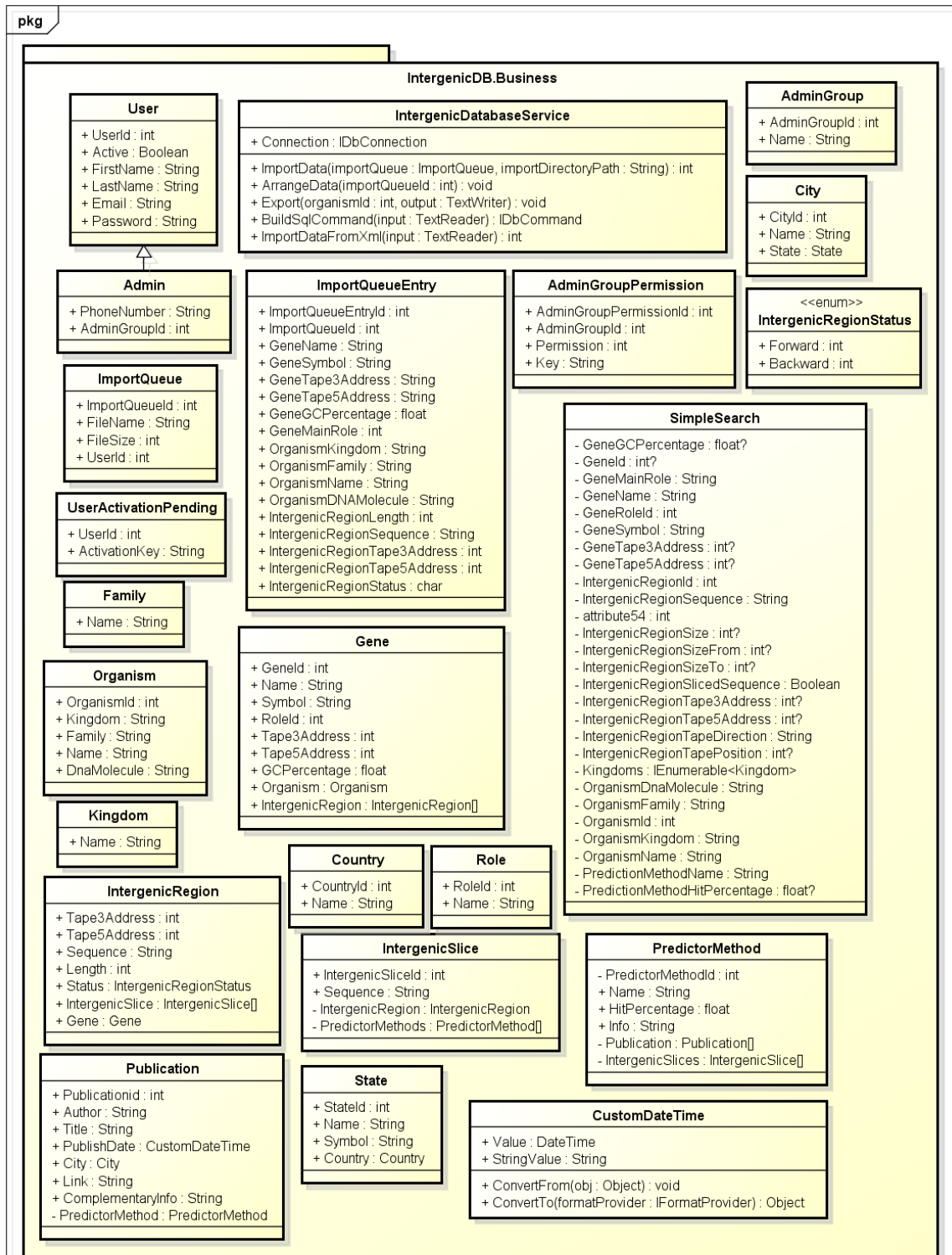
## ANEXO C – Diagrama de classes atualizado do projeto IntergenicDB.Data



powered by Astah

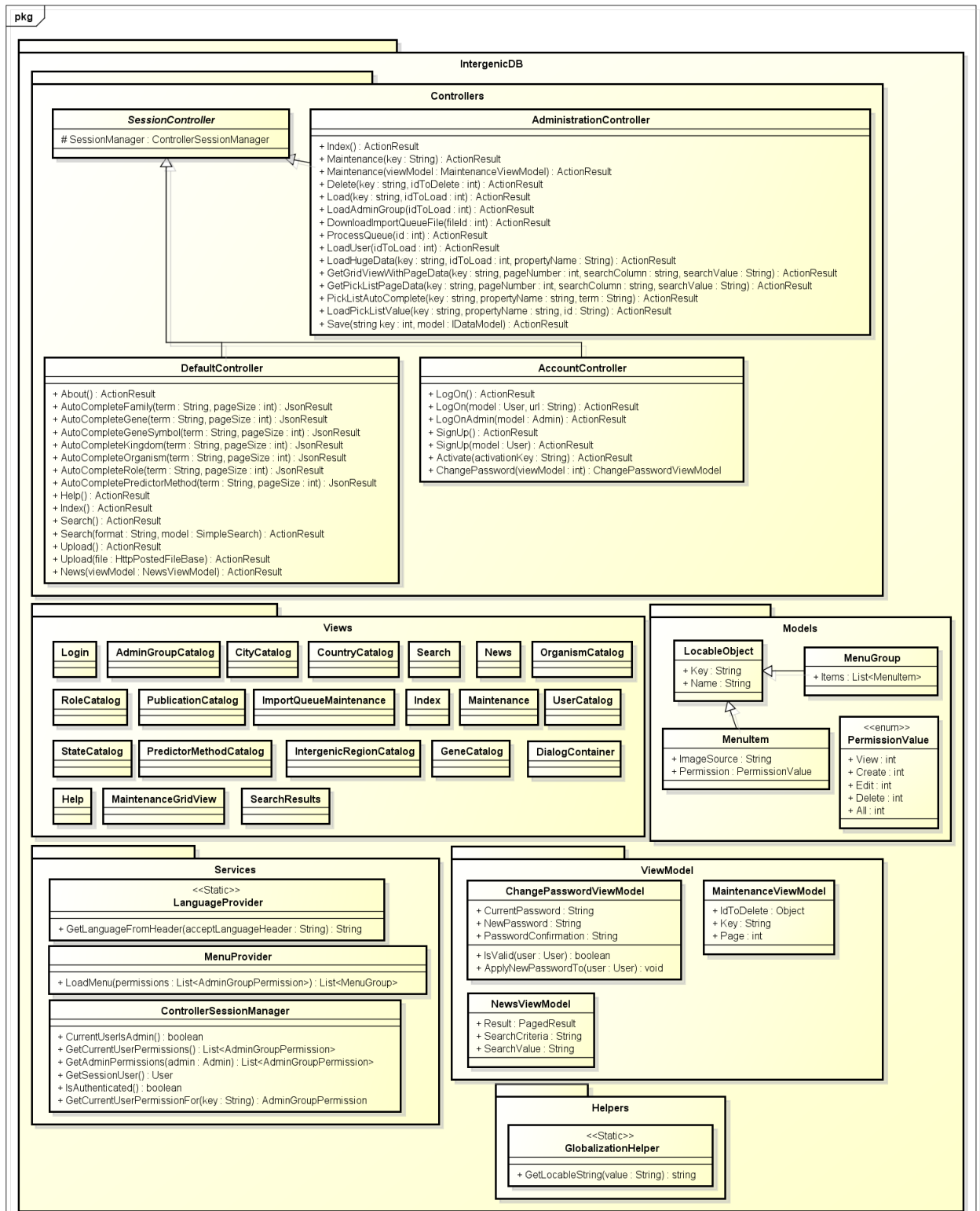


## ANEXO D – Diagrama de classes atualizado do projeto IntergenicDB.Business



powered by Astah

## ANEXO E – Diagrama de classes atualizado do projeto IntergenicDB



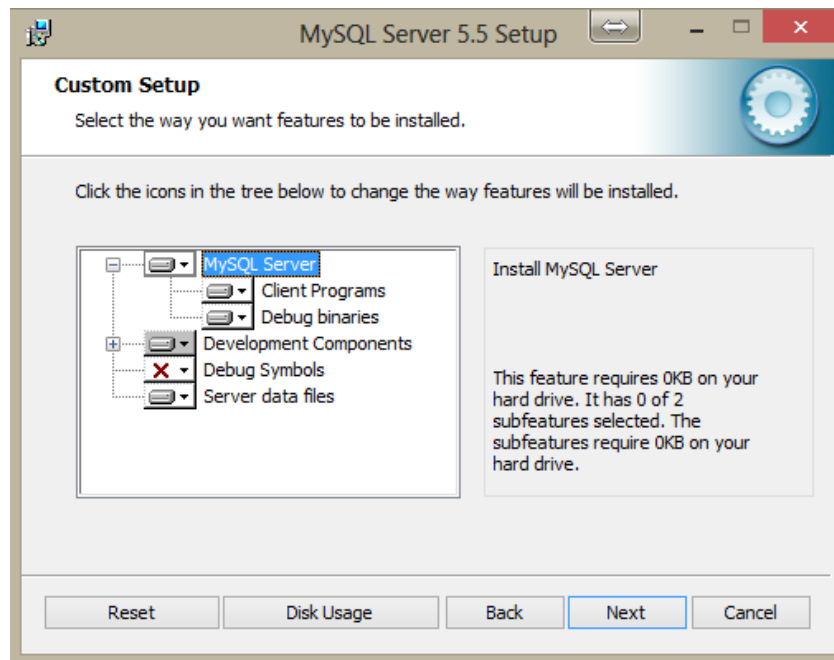
## ANEXO F – Processo de instalação do IntergenicDB

Neste anexo será demonstrado o processo de instalação do IntergenicDB em um servidor *web Microsoft Internet Information Services 8*.

### Instalação do *MySQL*

O armazenamento dos dados do IntergenicDB é efetuado em um banco *MySQL*, e por este motivo o mesmo deve estar instalado no servidor. A distribuição do *MySQL* que será utilizada é a *MySQL Community Server*<sup>70</sup> por ser totalmente gratuita.

Ao executar o instalador, será apresentado um *wizard*<sup>71</sup> de instalação que possibilitará informar os componentes que serão instalados e um usuário e senha para administração. Para a instalação de componentes pode ser utilizadas as configurações padrão existentes (Figura 68).



**Figura 68:** *Wizard* de instalação do *MySQL Community Server*

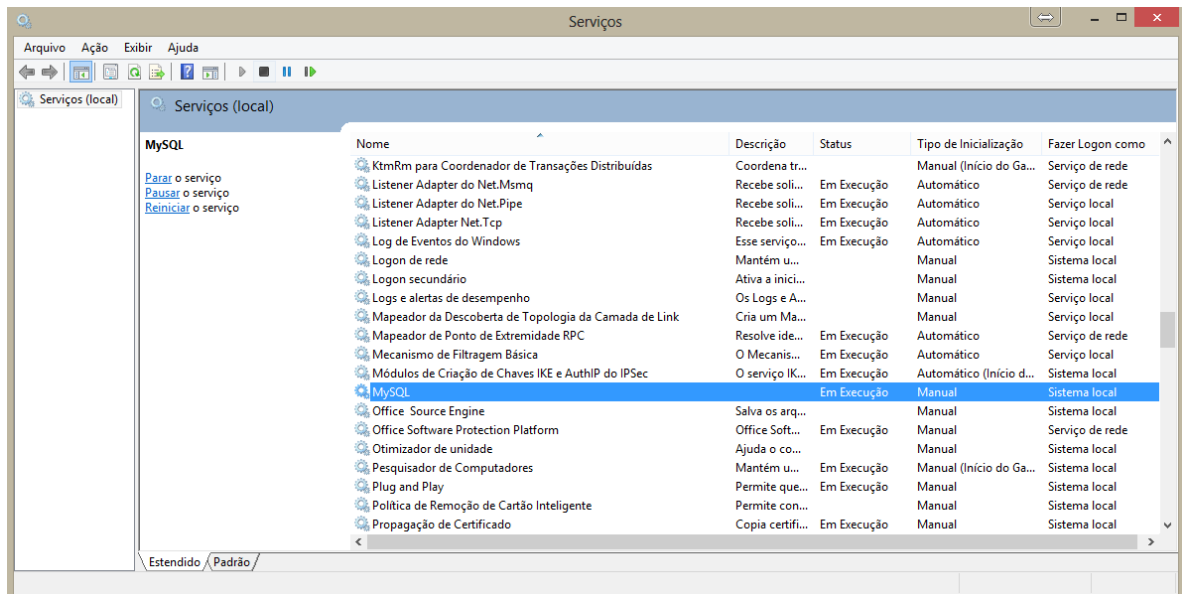
Ao clicar no botão *Next*, será exibida outra janela apenas solicitando a confirmação da instalação.

Quando instalado, o *MySQL* irá executar como um *Windows Service*<sup>72</sup> e estará disponível mesmo que o usuário não esteja autenticado no servidor (Figura 69).

<sup>70</sup> <http://www.mysql.com/downloads/mysql/>

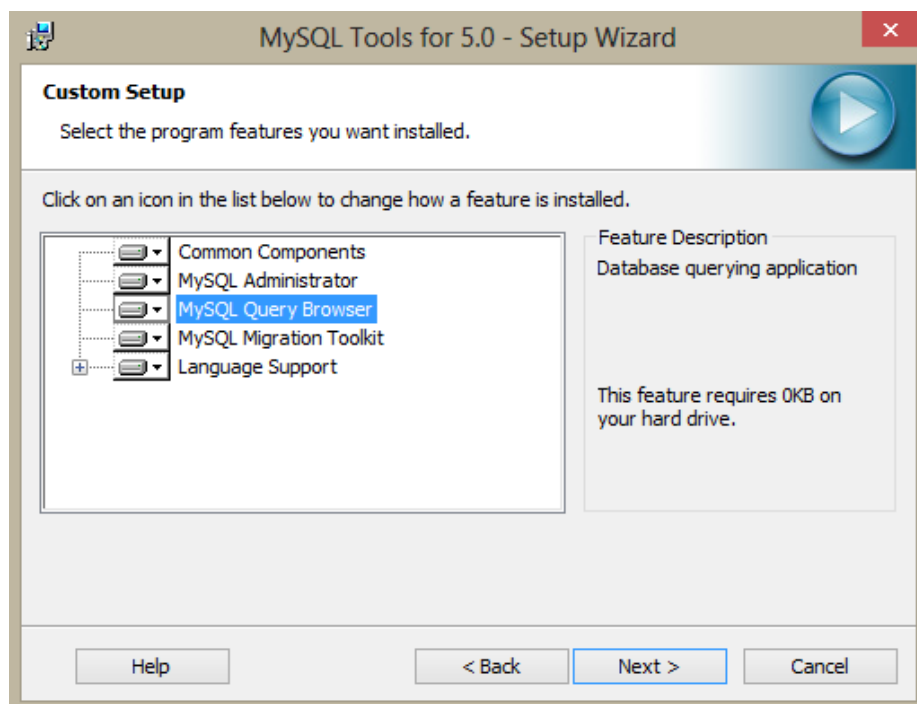
<sup>71</sup> *Wizard*: Ferramenta que utiliza um método passo-a-passo para efetuar a instalação de uma aplicação.

<sup>72</sup> *Windows Service*: Aplicações de longa duração que executam em suas próprias sessões do *Windows*, independente do usuário.



**Figura 69: Janela de serviços do windows (Windows Services)**

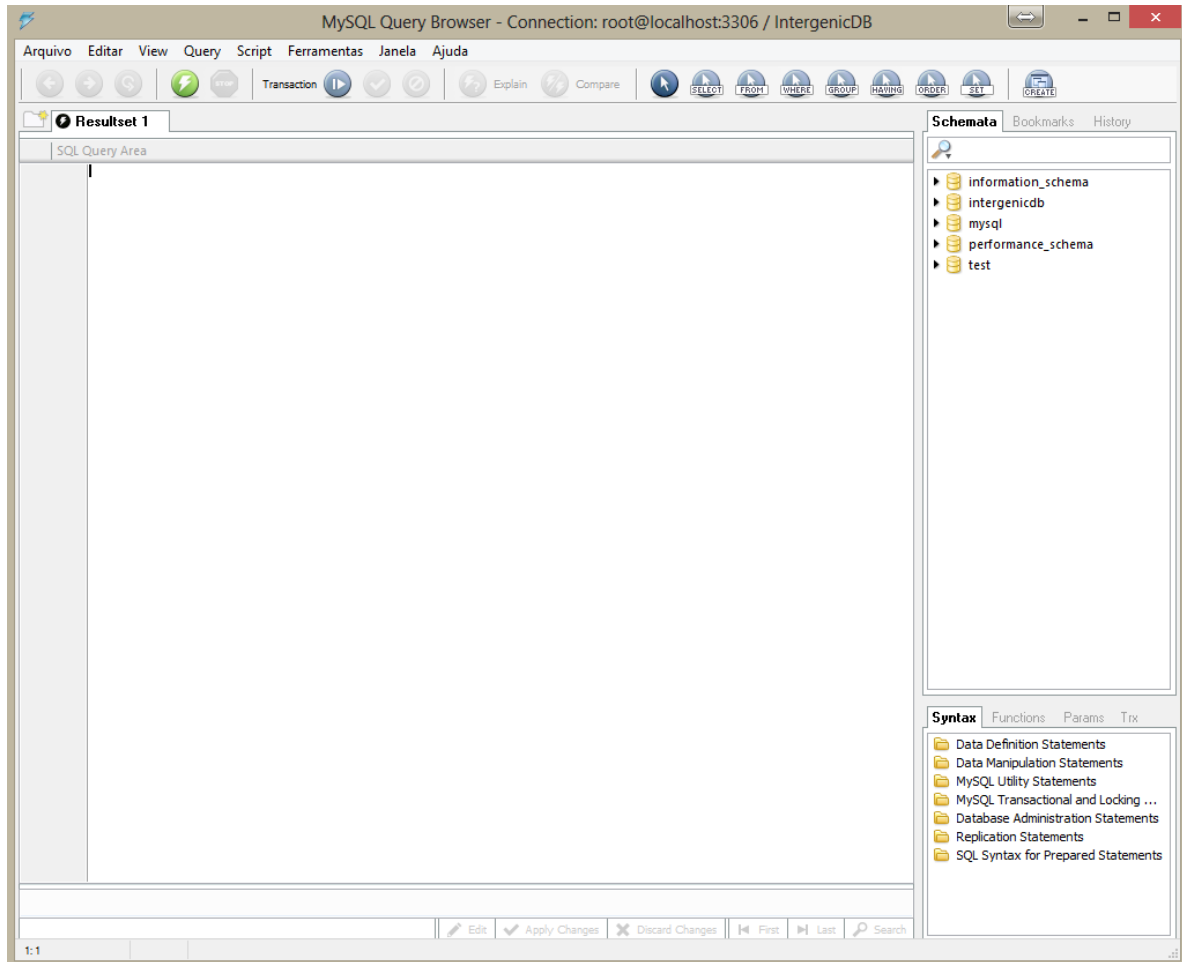
Neste momento apenas o servidor estará instalado, porém ainda é necessária uma ferramenta que permita administrar os bancos de dados no mesmo, utilizando linguagens DDL e DML. Para isto, é necessário instalar um pacote de ferramentas do MySQL chamado *MySQL GUI Tools*<sup>73</sup> (Figura 70).



**Figura 70: Instalação do MySQL GUI Tools**

<sup>73</sup> <http://dev.mysql.com/downloads/gui-tools/5.0.html>

Neste pacote está disponível a ferramenta *MySQL Query Browser*, que será utilizada para executar os Scripts de criação do banco de dados (Figura 71).

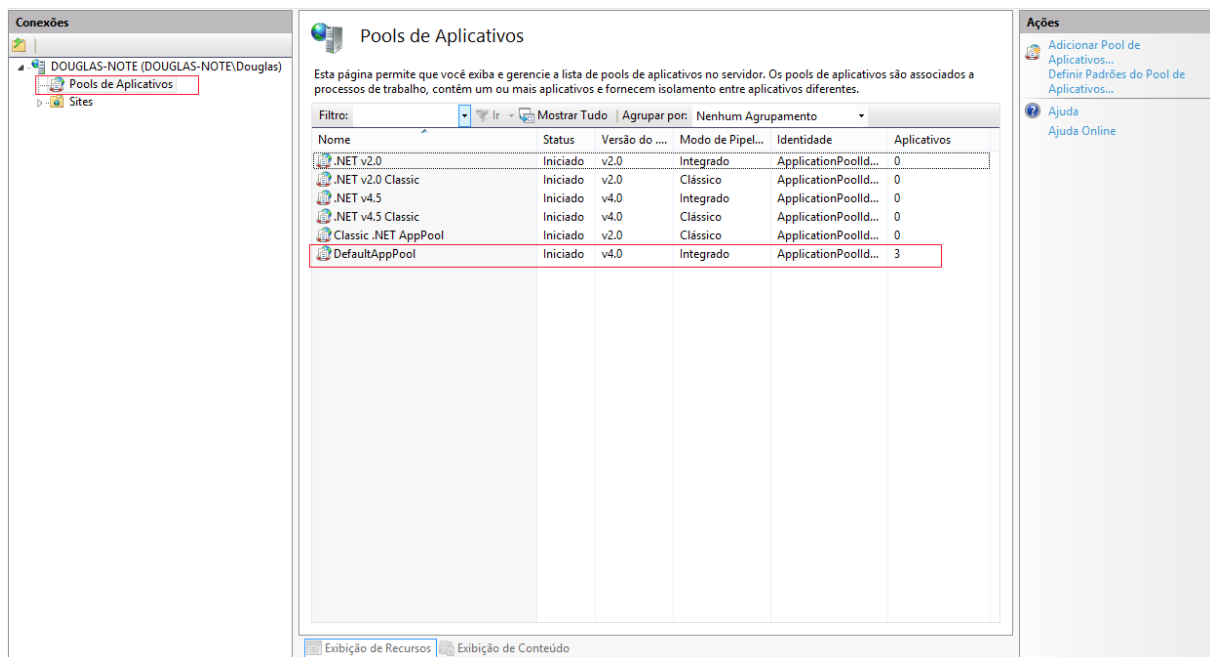


**Figura 71: *MySQL Query Browser***

Com esta ferramenta, é possível criar um banco de dados para o IntergenicDB, e executar os *scripts* para criar tabelas (*tables*), visões (*views*) e procedimentos armazenados (*stored procedures*). Ao final deste procedimento, o *MySQL* estará disponível para ser utilizado.

## Instalação da aplicação

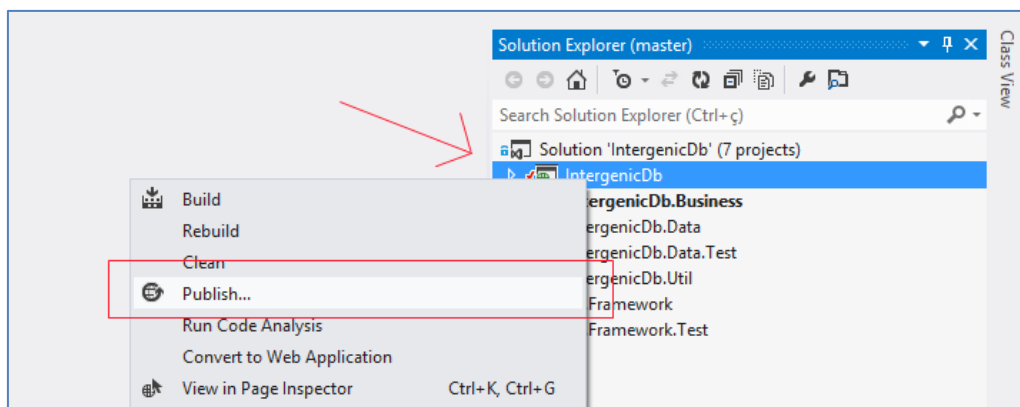
O primeiro passo garantir que a instalação ocorra com êxito, é verificar se algum *pool* de aplicativos do servidor está executando no modo de *pipeline* integrado<sup>74</sup> e a versão do *Microsoft .Net* existente é a 4.0 (Figura 72). Para isto, deve ser utilizado o gerenciador do *Internet Information Services* localizado nas Ferramentas Administrativas do *Windows*.



**Figura 72: Pools de aplicativos do IIS 8**

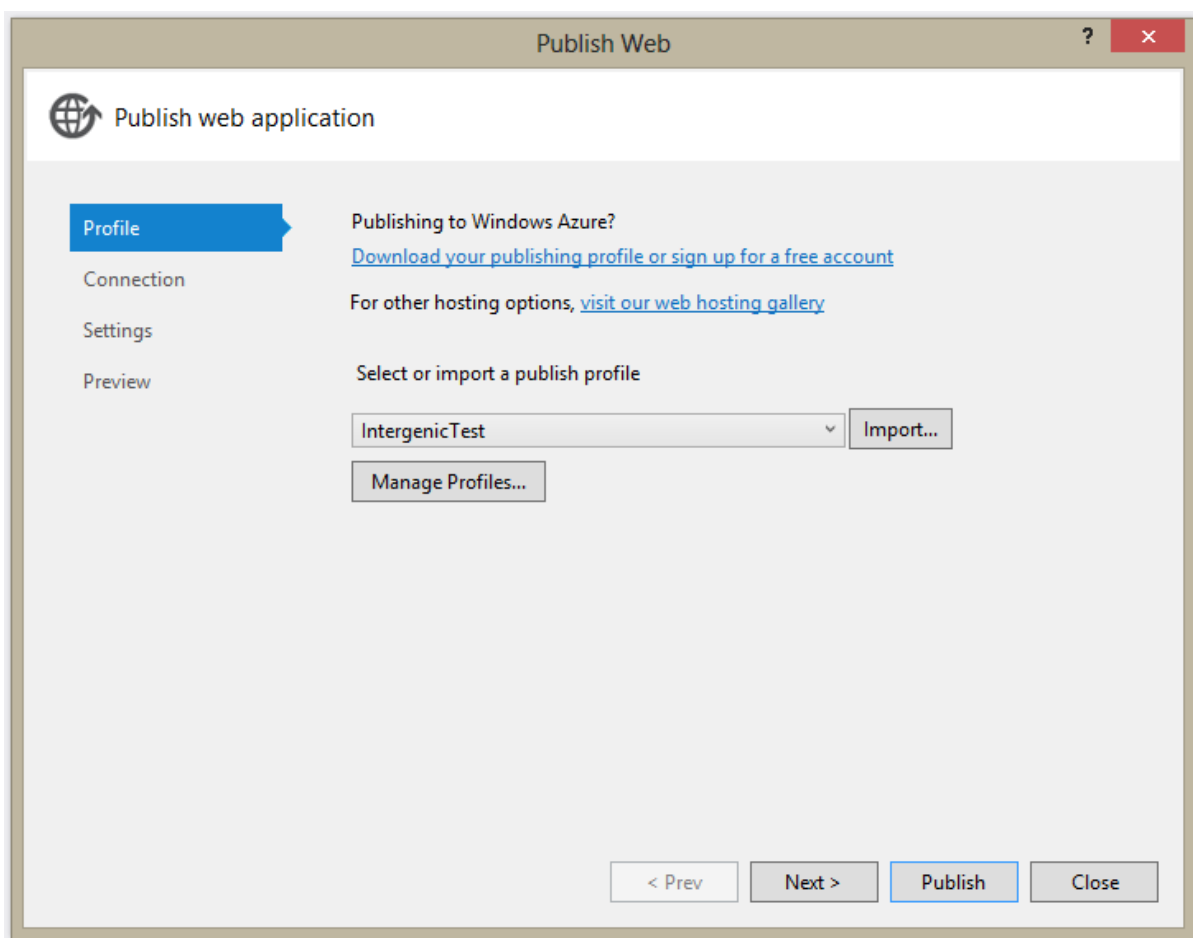
Em seguida, deve ser efetuada a publicação da website para a pasta *root* padrão do IIS (*c:\inetpub\wwwroot\*). Para isto, deve ser aberta a solução do IntergenicDB no Visual Studio e selecionado o projeto principal (IntergenicDB). Com o *mouse* sobre o projeto selecionado, deve ser clicado com o botão direito do *mouse* e selecionada a opção *Publish* (Figura 73).

<sup>74</sup> *Pipeline* Integrado: Habilita o ASP.Net a participar do processamento das requisições recebidas pelo IIS.



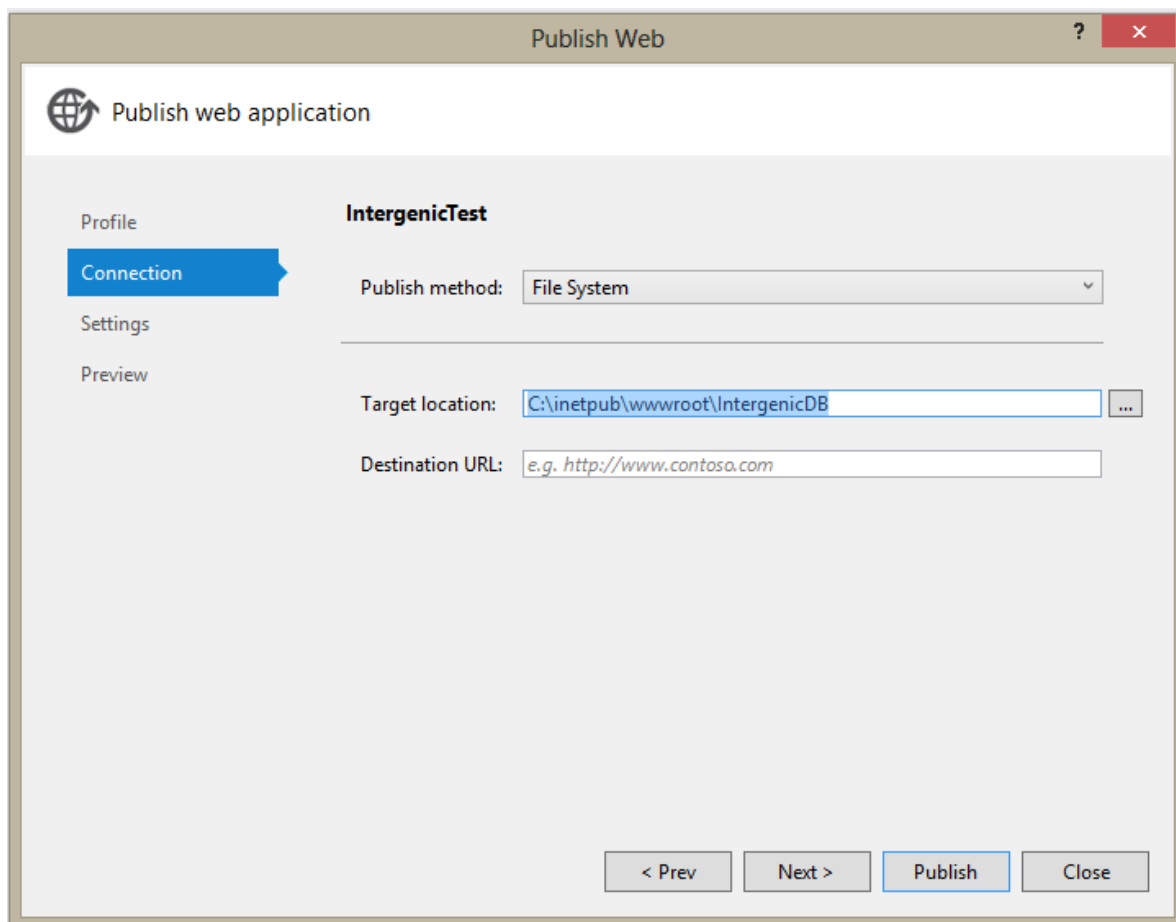
**Figura 73: Publicação no Microsoft Visual Studio**

Após selecionada a opção *Publish*, será aberta uma janela para configurar a publicação (Figura 74). Nesta janela, é possível selecionar um *profile* (utilizado para armazenar preferências de publicações anteriores), configurar a saída da publicação em *Connection*, efetuar as configurações de compilação da publicação em *Settings*, e observar um *Preview* da publicação.



**Figura 74: Configuração da publicação**

Acessando a opção *Connection*, será possível definir o método de publicação indicando se a mesma ocorrerá no sistema de arquivos do próprio computador, se será para um servidor FTP ou se será gerado um pacote para instalação no servidor *Web*. Para este exemplo de instalação, será utilizada a opção “*File System*” (Figura 75).



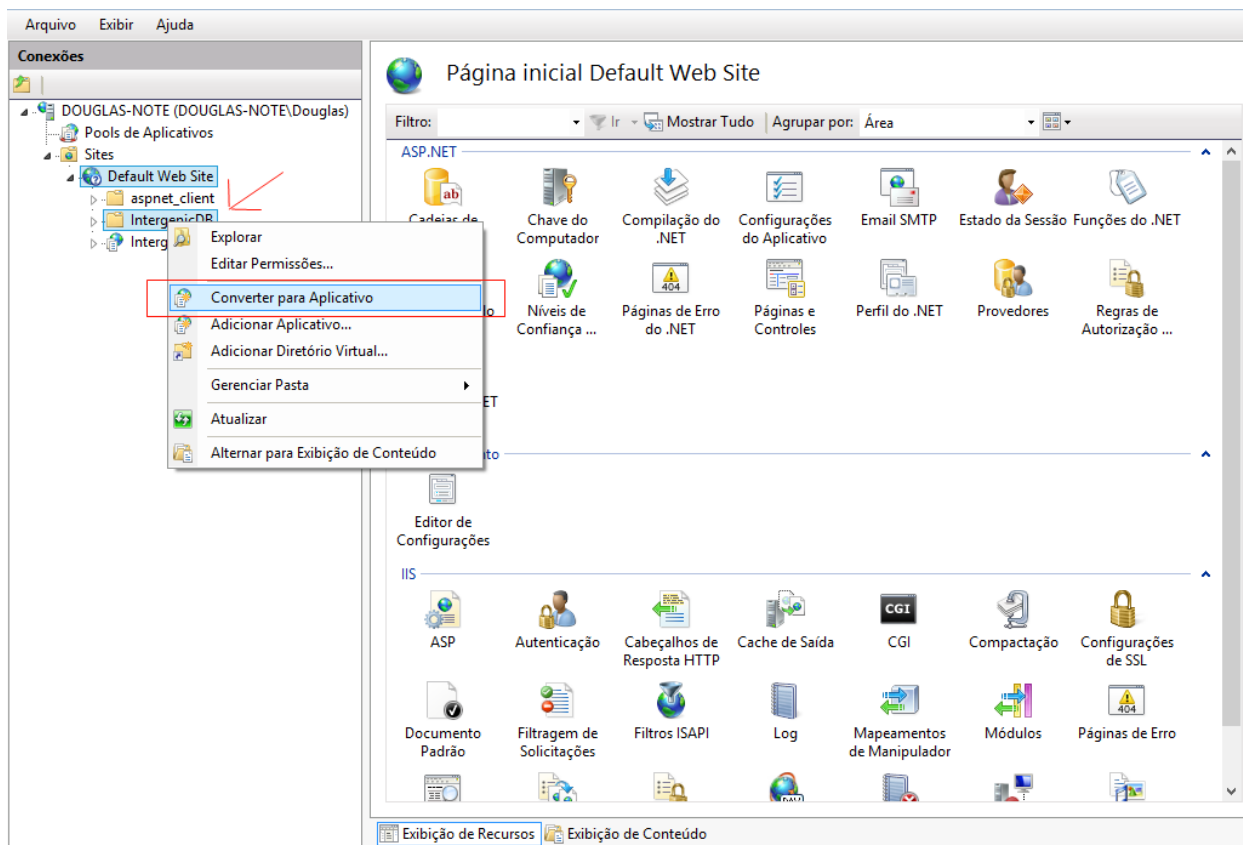
**Figura 75: Tela da opção *Connection***

Outro ponto importante é definir o campo *Configuration* como “*Release*” na opção *Settings*. Isto fará com que a compilação do produto para a publicação ocorra utilizando otimizações do compilador.

Depois de finalizada a configuração basta clicar no botão *Publish*. Isto fará com que o *Visual Studio* compile a aplicação e copie o resultado para a pasta informada em “*Target Location*”.

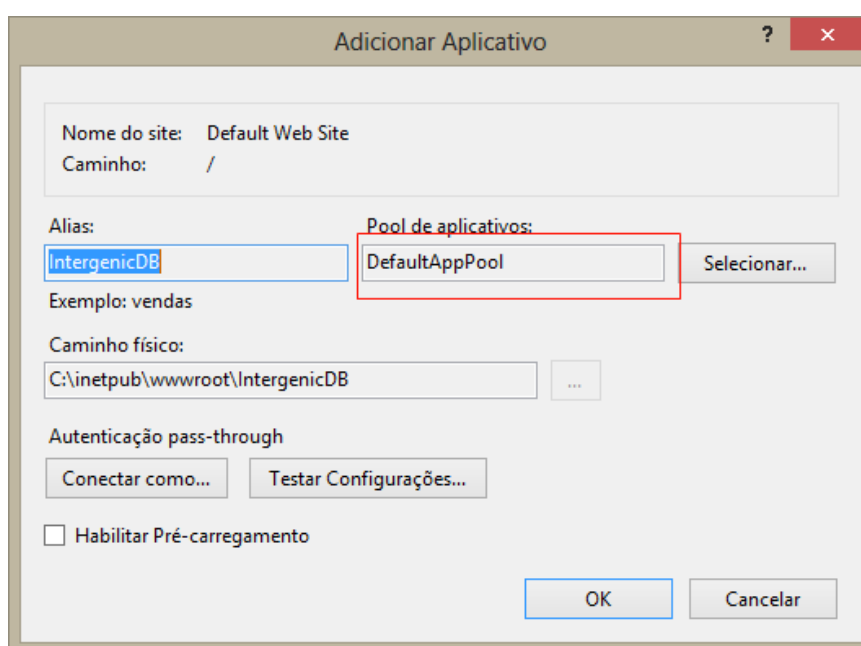
O último passo é transformar a pasta publicada em aplicação no IIS. Para efetuar este procedimento, é necessário selecionar a pasta publicada anteriormente na árvore de conexões do IIS, clicar com o botão direito na mesma e selecionar a opção “*Converter para Aplicativo*” (Figura 76).





**Figura 76: Converter aplicativo no IIS**

Durante a conversão do aplicativo, serão solicitadas algumas configurações para a conversão. É neste momento que o *pool* de aplicativos configurado anteriormente deverá ser selecionado (Figura 77).



**Figura 77: Configurações de Adição de Aplicativo**

O último passo a ser executado, é a configuração do banco de dados no IntergenicDB. Para isto, deve ser aberta o arquivo *Web.config*, existente na pasta publicada anteriormente, e incluir as informações necessárias para a conexão com o banco de dados (Figura 78).

```
<configuration>
  <connectionStrings>
    <add name="LP.Framework.Properties.Settings.ConnectionString"
      (A) connectionString="server=localhost;User Id=root;password=admin;
        Persist Security Info=True;database=intergenicdb;
        Allow User Variables=True;Use Procedure Bodies=false;"
        providerName="MySql.Data.MySqlClient" />
  </connectionStrings>

  <appSettings>...</appSettings>

  <system.web>...</system.web>

  <system.webServer>...</system.webServer>

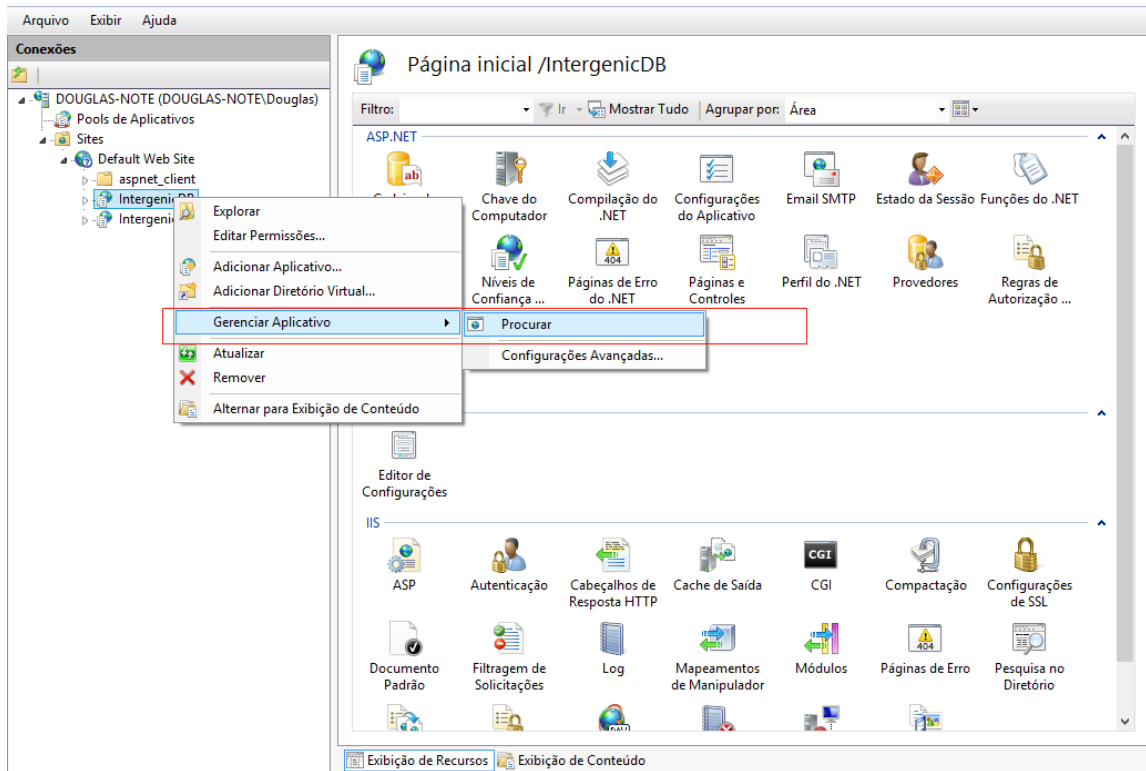
  <runtime>...</runtime>

  <location path="~/Default/Uploa">...</location>
</configuration>
```

**Figura 78: Configuração da conexão com o banco de dados**

A informação mais importante nesta configuração é a *connectionString* (A). Nela é informado o endereço do servidor, o identificador do usuário que será utilizado para a conexão, a senha deste usuário e o nome banco de dados que será conectado.

Para testar o resultado da instalação do IntergenicDB, deve ser selecionada a opção “Procurar” no gerenciador do IIS (Figura 79).



**Figura 79: Teste da instalação**