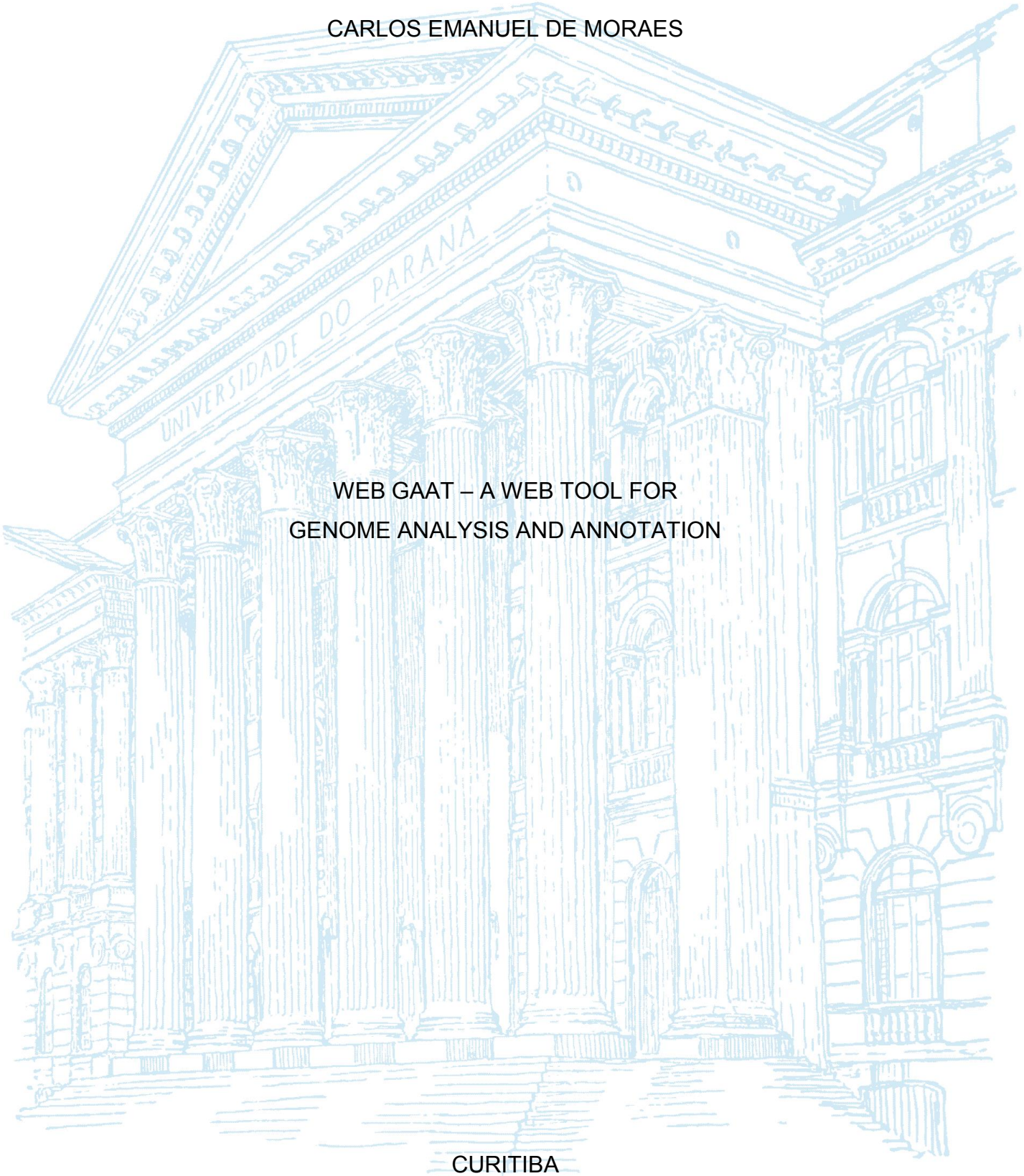
UNIVERSIDADE FEDERAL DO PARANÁ

CARLOS EMANUEL DE MORAES



WEB GAAT – A WEB TOOL FOR GENOME ANALYSIS AND ANNOTATION

CURITIBA

2017

CARLOS EMANUEL DE MORAES

WEB GAAT – A WEB TOOL FOR GENOME ANALYSIS AND ANNOTATION

Dissertação apresentada ao Curso de Pós-Graduação em Bioinformática, Área de Concentração em Bioinformática, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, como parte dos requisitos para obtenção do título de Mestre em Bioinformática.

Orientador: Prof. Dr. Alessandro Brawerman
Coorientador: Prof. Dr. Leonardo Magalhães Cruz

CURITIBA

2017

Catalogação na publicação
Sistema de Bibliotecas UFPR
Biblioteca de Educação Profissional e Tecnológica

M827 Moraes, Carlos Emanuel de
Web gaat – a web tool for genome analysis and annotation [recurso eletrônico] / Carlos Emanuel de Moraes. - Curitiba, 2017.
93 p.: il.

Dissertação (Mestrado) – Universidade Federal do Paraná, Setor de Educação Profissional e Tecnológica, Curso de Pós-Graduação em Bioinformática, 2017.
Orientador: Alessandro Brawerman
Coorientador: Leonardo Magalhães Cruz

1. Genomas. 2. Visualização da informação. 3. Bioinformática. I. Brawerman, Alessandro. II. Cruz, Leonardo Magalhães. II. Título. III. Universidade Federal do Paraná.

CDD 005.369

Elaboração: Angela Pereira de Farias Mengatto - CRB 9/1002

TERMO DE APROVAÇÃO

CARLOS EMANUEL DE MORAES

"WEB GAAT – A WEB TOOL FOR GENOME ANALYSIS AND ANNOTATION"

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre em Bioinformática, pelo Programa de Pós-graduação em Bioinformática, Setor de Educação Profissional e Tecnológica, da Universidade Federal do Paraná, pela seguinte banca examinadora:

Dr. Alessandro Brawerman
Programa de Pós-graduação em Bioinformática
Universidade Federal do Paraná - UFPR

Dr. Dieval Guizelini
Setor de Educação Profissional e Tecnológica
Universidade Federal do Paraná - UFPR

Dr. Rodrigo Luis Alves Cardoso
Bolsista PNPD/CAPES - Programa de Pós-graduação em Bioinformática
Universidade Federal do Paraná - UFPR

Curitiba, 15 de dezembro de 2016

AGRADECIMENTOS

Agradeço primeiramente a Deus e às pessoas que me inspiraram a seguir este caminho e não desistir, minha mãe, Ledi, minha esposa, Ana Carolina, minha filha, Cecília, e o orientador deste trabalho, Professor Alessandro Brawerman. Seu apoio foi fundamental para a conclusão deste percurso.

Agradeço ao coorientador, Professor Leonardo Magalhães Cruz, por suas contribuições e direcionamentos em momentos importantes do trabalho.

Agradeço aos colegas, professores e a todos os colaboradores do PPG-BIOINFO.

Por fim, agradeço à empresa Positivo Informática por ter permitido minha ausência em diversos momentos ao longo do curso.

“A ciência é, portanto, uma perversão de si mesma,
a menos que tenha como fim último, melhorar a humanidade.”
Nikola Tesla

RESUMO

A grande quantidade de dados genômicos, obtidos através de sequenciadores automáticos de DNA de nova geração, os NGS, tem possibilitado o estudo de genomas em níveis inéditos. Embora existam diferentes programas de computador direcionados às diversas tarefas relacionadas ao estudo destes dados, o complexo processo de análise e anotação de genomas frente ao variado volume de informação tem sido um fator limitante em inúmeros casos. Não só os programas de computador utilizados precisam oferecer um processamento eficiente como precisam também ser intuitivos e de fácil utilização. Neste aspecto, a Visualização da Informação (VI), a qual se apoia nos limites da visão e percepção humana, para desenvolver métodos computacionais eficazes para interação e compreensão de dados em grande escala, se mostra como um diferencial para obtenção de melhores resultados. Após analisar as diferentes ferramentas disponíveis, suas interfaces, utilização e resultados gerados, este trabalho apresenta o desenvolvimento do Web GAAT, uma plataforma que tem como objetivo reunir os principais programas utilizados para análise e anotação de genomas em uma interface única, que faz uso de recursos de VI, possibilitando a fácil visualização e interação com genomas de procariotos, em projetos individuais ou colaborativos.

Palavras-chave: Visualização de genomas. Anotação de genomas. Visualização da informação.

ABSTRACT

The large amount of genomic data obtained through next-generation sequencing, the NGS, has allowed the study of genomes in unprecedented levels. Although there are many computer programs targeted to the various tasks related to the study of these data, the complex process of genome analysis and annotation facing the vast amount of information has been a limiting factor in many cases. Computer programs not only must offer efficient processing but must also be intuitive and easy to use. In this aspect, the Information Visualization (VI), which rests within the limits of human perception and vision to develop effective computational methods for interaction and understanding of large-scale data, can possibly be a differentiator for obtaining better results. By analyzing the different tools available, their interfaces, use and outputs, this paper presents the development of Web GAAT, a platform that aims to bring together the main programs used in genome analysis and annotation. Under a single interface, based on VI technics, the platform provides easy visualization and interaction with prokaryotic genomes, in individual or collaborative projects.

Keywords: Genome visualization. Genome annotation. Information visualization.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

BLAST	-	Basic Local Alignment Search Tool
DNA	-	Ácido desoxirribonucleico
EMBL	-	European Molecular Biology Laboratory
ENA	-	European Nucleotide Archive
GAAT	-	Genome Assembly and Analysis Tool
GMOD	-	Generic Model Organism Database
INSDC	-	International Nucleotide Sequence Database Collaboration
MVC	-	Model View Controller
mRNA	-	RNA mensageiro
NCBI	-	National Center of Biotechnology Information
NGS	-	Next-Generation Sequencing
NIG	-	National Institute of Genetics
ORF	-	Open Reading Frame
rRNA	-	RNA ribossomal
SGBD	-	Sistema Gerenciador de Banco de Dados
tRNA	-	RNA de transcrição
UI	-	User Interface (Interface do Usuário)
VI	-	Visualização da Informação
VM	-	Virtual Machine (Máquina Virtual)
Web GAAT	-	Web Genome Analysis and Annotation Tool

SUMÁRIO

1 INTRODUÇÃO

A grande quantidade de dados genômicos, obtidos através de sequenciadores automáticos de DNA de nova geração (NGS, do inglês *Next-Generation Sequencing*), tem permitido aos pesquisadores estudar sistemas biológicos em um nível nunca antes possível. Embora existam diversas ferramentas disponíveis para auxiliar o complexo processo de análise e anotação destes dados, o vasto e variado volume de informação tem sido um fator limitante em inúmeros estudos (THORVALDSDÓTTIT et al., 2013). Para minimizar essa limitação, diversos programas de computador foram desenvolvidos, porém fatores como dependência de sistema operacional ou de determinada tecnologia, falta de uma interface gráfica amigável acabam, muitas vezes, comprometendo o desempenho de uma pesquisa. Os programas de computador utilizados precisam fornecer, não somente um processamento de dados eficiente, mas também, interfaces gráficas intuitivas com uma visualização da informação (VI) que permita ao pesquisador explorar e interagir com a diversidade de dados biológicos existentes com maior facilidade, objetivando-se puramente à pesquisa.

Por séculos a visualização da informação tem auxiliado o homem a compreender com maior facilidade dados dos mais variados tipos (CARD et al, 1999). No campo da genômica, técnicas de VI, baseadas nos limites de visão e percepção humana, são indispensáveis para a visualização de grandes conjuntos de dados, pois possibilitam interação e entendimento de informações em tarefas que seriam praticamente impossíveis de serem realizadas a olho nu (TAO et al., 2004).

Neste cenário, explorando diferentes programas para análise e anotação de dados genômicos, com os diferentes tipos de resultados originados por estes programas, foi identificada uma oportunidade para desenvolvimento de um software Web que reúna os principais recursos necessários para análise e anotação de dados sequenciados, fornecendo uma interface gráfica única para visualização da informação e interação, possibilitando maior eficiência no estudo de dados sequenciados.

Nomeado Web GAAT, um acrônimo para *Genome Analysis and Annotation Tool* (Ferramenta para Análise e Anotação de Genomas), precedida

pelo termo Web por se tratar de um software Web, o presente trabalho busca a disponibilização de uma plataforma que reúna em uma interface única recursos para visualização e anotação de genomas de procariotos, como a identificação automática de ORFs, sequências de DNA que podem codificar proteínas, a possibilidade de anotar dados sobre estas ORFs, a visualização e interação com genomas parciais ou completos utilizando técnicas de VI, e o compartilhamento de projetos entre pesquisadores a fim de dividir o esforço durante o processo de análise e anotação.

O trabalho propõe também que a plataforma possibilite a adição de novos recursos, bem como fácil atualização ou remoção de funcionalidades existentes. Desta forma, novas ferramentas podem ser integradas, aumentando a vida útil do programa.

Por fim, o resultado deste trabalho busca a disponibilização de uma nova ferramenta que substitua o programa homônimo GAAT *(Genome Assembly and Analysis Tool)*, desenvolvido e mantido pelo Laboratório de Bioinformática do Núcleo da Fixação Biológica de Nitrogênio da UFPR (TIEPPO, 2011).

1.1 OBJETIVOS

Esta seção apresenta os objetivos, tanto geral como específicos, do projeto, buscando estabelecer o que se espera conquistar com o desenvolvimento do mesmo.

1.1.1 Objetivo Geral

O trabalho tem como objetivo o desenvolvimento de uma plataforma Web capaz que possibilite ao usuário visualizar, interagir e anotar dados genômicos sequenciados.

1.1.2 Objetivos Específicos

Os objetivos específicos foram baseados no conjunto de funcionalidades que o programa atende:

a) Identificação automática de ORFs, tRNAs e predição de genes;

b) Possibilidade de realizar anotação sobre dados obtidos;
c) Conversão automática de sequências de nucleotídeos em sequência de proteínas;
d) Possibilidade de envio de sequências para Web BLASTN e Web BLASTP;
e) Visualização de genomas parciais e completos;
f) Possibilidade de criação de projetos individuais;
g) Possibilidade de compartilhamento de projetos para que mais de um pesquisador tenha acesso aos dados;
h) Armazenamento de histórico sobre dados anotados;
i) Arquitetura flexível para adição de novas funcionalidades;
j) Código de fácil atualização e manutenção.
k) Substituição do programa homônimo GAAT.

1.2 JUSTIFICATIVA E RELEVÂNCIA DO TRABALHO

A quantidade de dados genômicos obtidos através dos NGS, sequenciadores automáticos de DNA de nova geração, trouxe consigo a demanda por programas de computador eficientes, capazes de lidar com o vasto volume de informação, auxiliando os pesquisadores nas diversas atividades do estudo de genomas.

Embora inúmeros programas de computador, como o GLIMMER (SALZBERG et al., 1998), programa destinado à predição de genes, Artemis (RUTHERFORD et al., 2000; SANGER, 2014) amplamente utilizado para visualização de sequencias e BLAST (ALTSCHUL et al, 1990; NCBI, 2016), destinado à comparação genômica, tenham sido desenvolvidos com o objetivo de melhorar os avanços no estudo destes dados, a utilização destes pode ser trabalhosa. Dependências de sistema operacional ou de outros recursos, algumas vezes não familiares aos pesquisadores e diferentes interfaces podem reduzir a produtividade, comprometendo os resultados.

Neste cenário, é relevante o desenvolvimento do Web GAAT, uma vez que o programa visa integrar os principais recursos para análise e anotação de genomas em uma interface única e on-line, dispensando a necessidade de instalação de diferentes programas e possibilitando ainda o trabalho

colaborativo, para que mais de um pesquisador possa trabalhar no mesmo conjunto de informações.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

Os próximos capítulos deste documento têm o objetivo de apresentar o desenvolvimento deste trabalho. Este documento está dividido da seguinte forma, o Capítulo 2 apresenta uma revisão bibliográfica que visa auxiliar no entendimento dos principais conceitos e tecnologias que fundamentaram este trabalho. O Capítulo 3, por sua vez, apresenta a metodologia utilizada para o desenvolvimento do Web GAAT e seus recursos. O Capítulo 4 apresenta os testes e experimentos práticos executados, bem como os resultados obtidos. Por fim, o Capítulo 5 conclui o trabalho e apresenta as possibilidades para desenvolvimento futuro.

2 REVISÃO DA LITERATURA

Este capítulo destina-se a apresentar conceitos importantes e fornecer fundamentação teórica para um melhor entendimento do trabalho.

Inicialmente são apresentados os conceitos em torno da análise e anotação de genomas, justificando sua importância. Em seguida são relatados os principais trabalhos relacionados. Ao final é apresentada uma visão geral sobre as principais tecnologias envolvidas no desenvolvimento de sistemas para Web.

2.1 MONTAGEM E ANOTAÇÃO DE GENOMAS

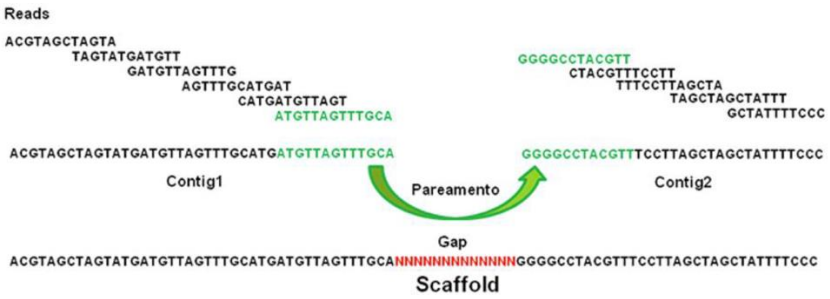
Com o advento dos sequenciadores automáticos de DNA de nova geração, conhecidos como NGS, a quantidade de dados genômicos obtidos possibilitou o estudo de sistemas biológicos em níveis até então inéditos (THORVALDSDÓTTIT et al., 2013). Significativamente mais baratos, mais rápidos e mais precisos, os NGS possibilitam obter sequências a partir de apenas uma amostra de DNA, enquanto que para o método de Sanger, para a construção de uma sequência de 100 pares de base (bp), centenas de cópias da mesma amostra são necessárias (EBI, 2012).

Nos dois cenários o sequenciamento é realizado de maneira aleatória e programas computacionais são utilizados para fazer a sobreposição de sequências (*reads*) em um processo chamado de montagem (*assembly*) com o objetivo de obter uma sequência de bases contíguas (*contigs*) (ZAHA et al., 2014).

Cardoso (2015) explica que os programas computacionais utilizados na montagem de genomas assumem que pares de *reads* com regiões sobrepostas ocorrem em uma mesma região do genoma, podendo assim ser utilizados para estender sua sequência de bases.

Conforme observado na Figura 1, *reads* com sobreposição são unidos em contigs que, por sua vez, existindo pareamento, podem ser ordenados em scaffolds. Às sequencias ausentes é atribuído o nome de gap (CARDOSO, 2015).

FIGURA 1 - SOBREPOSIÇÃO DE SEQUÊNCIA DE BASES



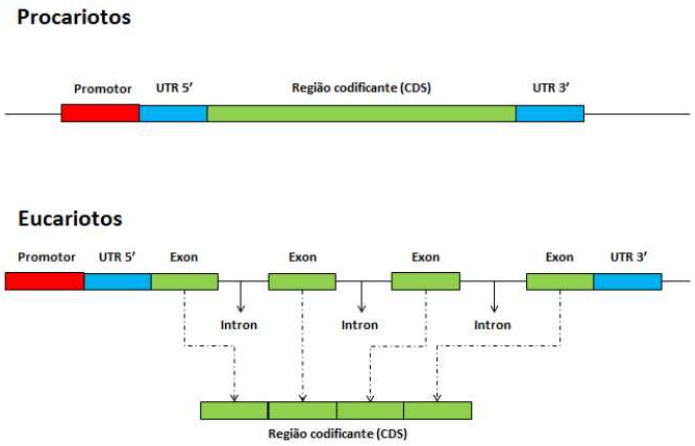
FONTE: Cardoso, 2015.

Após a montagem de um genoma, é realizada a anotação, processo que consiste na atribuição de um significado biológico às sequencias genômicas (ZAHA et al., 2014). A identificação e a caracterização destas regiões funcionais, o que pode incluir genes, promotores, terminadores, dentre outras, sendo estas usualmente denominadas *features,* pode ser realizada através de dados experimentais, como o alinhamento de sequências de transcritos ou proteínas, ou com base em ferramentas automatizadas de predição (KREMER; PINTO, 2016).

A utilização de ferramentas automatizadas de predição pode ainda ser comparada com a definição de (KOONIN; GALPERIN, 2003), na qual o processo de anotação é um subcampo da análise geral de genomas e qualquer tratamento de sequências genômicas pode ser realizado por meios computacionais.

A identificação de regiões funcionais de um genoma varia de acordo com o organismo estudado. Em procariotos, a porção codificante do gene é contida em uma única região. No caso de eucariotos, a presença de sequências intervenientes denominadas *introns*, interrompe a região transcrita do gene e a divide em vários segmentos, denominados *exons* (ZAHA et al., 2014). Essas diferenças podem ser observadas na Figura 2, na qual é possível observar que genes procariotos possuem estrutura mais simples, uma vez que não possuem regiões intermediárias (*introns*) entre regiões codificantes (*exons*).

FIGURA 2 - ESTRUTURA GÊNICA DE PROCARIOTOS E EUCARIOTOS



FONTE: Kremer; Pinto, 2016

A identificação de *introns* dentro de uma sequência gênica é extremamente complexa e, embora existam padrões de sequência que indicam a presença de uma junção entre *exons* e *introns*, estas podem variar. Desta forma, a utilização de ferramentas computacionais de predição em eucariotos é geralmente acompanhada do uso de dados experimentais (KREMER; PINTO, 2016).

Em procariotos, por outro lado, a ausência de *introns* possibilita que programas computacionais identifiquem códons de iniciação de proteínas (trincas de nucleotídeos geralmente iniciando com ATG) e códons de terminação na mesma fase de leitura (ZAHA et al., 2014).

Após a obtenção de sequências na fase de montagem ocorre a predição de genes, onde o programa de computador comumente mais utilizado para a identificação de regiões codificantes, em procariotos, é o GLIMMER, do inglês "*Gene Locator and Interpolated Markov Modeler*" (CARDOSO, 2015). O programa caracteriza-se por utilizar um conjunto de sequências de referência para treinar um modelo de predição que pode então ser utilizado para encontrar genes em meio ao genoma estudado (SALZBERG et al, 1998).

No caso de eucariotos, por serem mais complexos, muitas vezes são utilizados dados experimentais para suportar os modelos preditos computacionalmente.

Ocorre também na fase de predição a busca por ORFs, do inglês *Open Reading Frame*, que consistem em sequências ininterruptas de códons que

potencialmente podem codificar uma proteína. Estas são usadas como evidência para auxiliar na predição de genes e se caracterizam por iniciar a partir de um códon de início e terminar em um códon de finalização (NCBI, 2016). O ORF Finder (NCBI, 2016) é um dos programas utilizados com esta finalidade.

O próximo passo no processo de anotação, após a predição de genes, consiste em atribuir funções às proteínas obtidas. Para isso são usadas ferramentas de alinhamento, como o BLAST (ALTSCHUL et al, 1990; NCBI, 2016), do inglês "*Basic Local Alignment Search Tool*", que busca por proteínas similares à proteína de interesse em bancos de dados biológicos. O processo também é realizado a fim de eliminar falsas ORFs (STEIN, 2001; YANDELL & ENCE, 2012; CARDOSO, 2015).

Enquanto o processo de anotação visa principalmente a predição de regiões codificadoras de proteínas, programas específicos podem ser utilizados para encontrar outros elementos presentes na sequência, como é o caso do tRNA-scan SE, utilizado para predição de RNAs transportadores (tRNAs) (CARDOSO, 2015; KREMER & PINTO, 2016).

Por fim, uma vez que os elementos existentes no genoma são identificados, os dados anotados podem ser disponibilizados publicamente em bancos de dados biológicos, fornecendo um recurso essencial para outros projetos de anotação de genomas, constituindo as operações do dia-a-dia da biologia molecular (YANDELL & ENCE, 2012).

2.2 FERRAMENTAS PARA ANOTAÇÃO DE GENOMAS

Esta sessão tem como objetivo analisar mais especificamente os programas computacionais relatados no capítulo anterior a fim de identificar quais podem ser integrados ao Web GAAT e disponibilizados em uma mesma interface para utilização, conforme proposta da plataforma.

Os seguintes critérios foram utilizados durante a análise:

- Funcionalidade: A que o programa se destina;
- Formato de entrada: Como as sequências genômicas podem ser lidas pelos programas;
- Formato de saída de dados: Como são gerados os resultados do programa;

- Possibilidade de integração: Se o programa pode ser ou não integrado em sistemas Web.

A seguir estão descritos os softwares analisados.

2.2.1 ORF Finder

O ORF Finder (*Open Reading Frame Finder*), é uma ferramenta on-line de análise gráfica que possibilita a busca por ORFs em cada uma das seis fases de leitura de uma sequência de DNA (NCBI, 2016).

Após informada uma sequência no formato FASTA, o programa exibe as ORFs encontradas, seu tamanho, sua respectiva sequência de proteínas e possibilita ainda que sejam validadas, a partir do envio das mesmas para os programas SMART BLAST ou BLASTP, onde podem ser comparadas com outras sequências.

Uma vez que a utilização on-line do ORF Finder é limitada a arquivos com no máximo 50kb, é possível instalar e utilizar a ferramenta localmente em máquinas Linux de 64 bits. Embora exista essa possibilidade, não é possível integra-la com outros programas, pois a utilização se limita à interface do próprio programa.

2.2.2 GLIMMER

GLIMMER, do inglês "*Gene Locator and Interpolated Markov Modeler*", é um programa destinado à predição de genes. O programa utiliza um conjunto de sequências de referência para treinar um modelo de predição (que são padrões obtidos através do modelo oculto de Markov) e então esse modelo é aplicado a uma sequência informada pelo usuário a partir de um arquivo no formato FASTA (SALZBERG et al, 1998).

Diferente do ORF Finder, não possui interface gráfica para utilização, sendo executado por linhas de comando, mas seus resultados são gerados em arquivos de texto, podendo estes serem utilizados por outros programas a partir da criação de interpretadores (*parsers*) para extrair os dados presentes nos arquivos gerados pelo programa.

2.2.3 tRNAscan-SE

O tRNAscan-SE é um programa escrito em Perl utilizado para encontrar tRNAs em sequências de DNA. Combinando três métodos de pesquisa, o programa possui um índice de acerto entre 99 e 100%, resultando menos de um falso-positivo por quinze bilhões de nucleotídeos analisados em uma sequência (LOWE; EDDY, 1997).

Tendo como entrada uma sequência no formato Fasta, o programa é executado através de linhas de comando, ou seja, não possui interface gráfica para interação dos usuários. Também dispensa instalação, sendo necessário apenas que o computador onde o programa será disponibilizado tenha suporte a execução de scripts em Perl.

Como os resultados gerados pelo programa são arquivos de texto, a integração com outros softwares é possível a partir da utilização de *parsers* para a extração dos dados presentes nestes arquivos.

2.2.4 BLAST

O BLAST, um acrônimo para *Basic Local Alignment Search Tool*, é uma ferramenta utilizada para comparação de similaridade entre sequências. O programa compara sequências locais de nucleotídeos ou proteínas com dados armazenados em bancos de dados biológicos e calcula a significância estatística dos resultados. Pode ser utilizado para inferir relações funcionais e evolutivas entre sequências, bem como ajudar a identificar os membros de famílias de genes (ALTSCHUL, 1990; NCBI, 2016).

Principalmente utilizado a partir de seu site, com uma interface Web, possibilita que o usuário escolha as seguintes opções para comparação de sequências:

- BLASTN: Opção para comparação entre uma sequência de nucleotídeos e uma base de nucleotídeos;
- BLASTP: Opção para comparação entre uma sequência de proteínas e uma base de proteínas;
- BLASTX: Opção para comparação com uma base de proteínas a partir de uma sequência de nucleotídeos traduzida;

- TBLASTN: Opção para comparação entre uma sequência de proteínas e uma base traduzida de nucleotídeos;
- TBLASTX: Opção para comparação entre uma sequência traduzida de nucleotídeos e uma base traduzida de nucleotídeos.

Para utilização deve-se informar uma sequência no formato FASTA ou um código, chamado de *Accession Number*, ou número de acesso, que é um identificador único atribuído para uma sequência de DNA ou sequência de proteínas a fim de diferencia-los.

Os resultados regrados pelo programa são apresentados em sua própria interface, mas podem ser baixados em formatos mais simples, podendo ser interpretados por outros programas, possibilitando integração.

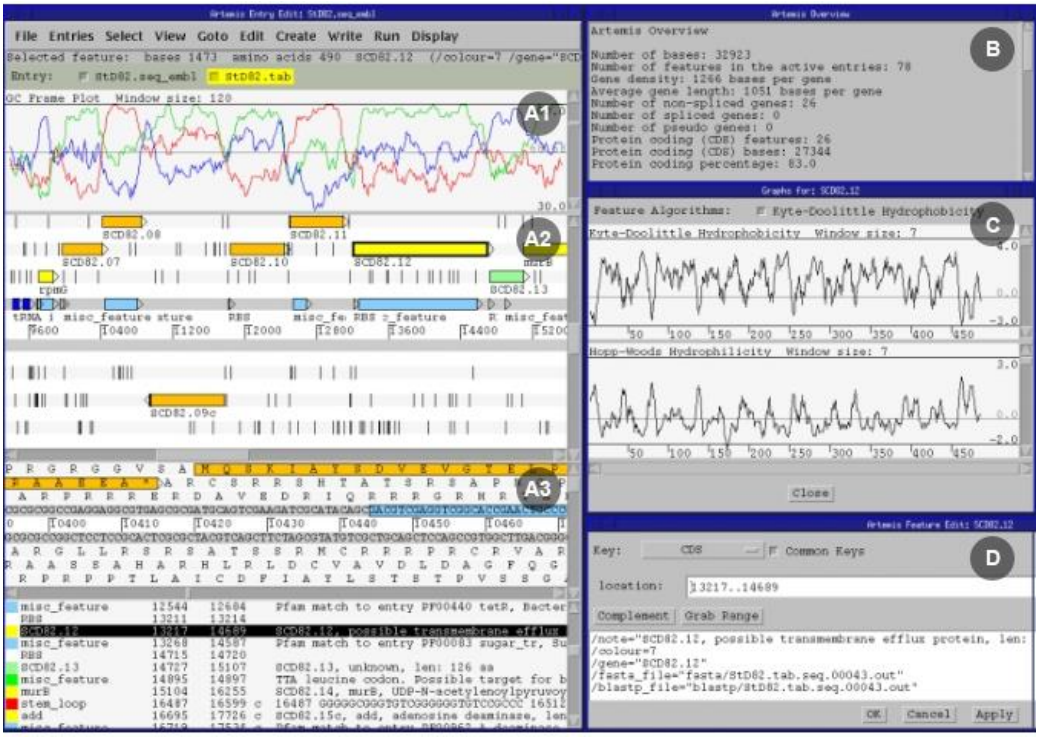
O BLAST também pode ser instalado localmente, possibilitando comparações com bases de dados locais ou com as bases padrão de pesquisa disponibilizadas pelo programa. Neste caso não há interface, a execução do programa é realizada a partir de linhas de comando e os resultados obtidos são gravados em arquivos de texto.

2.2.5 Artemis

O programa Artemis é uma ferramenta para visualização de sequências *(genome browser)* e anotação que possibilita que quaisquer dados provenientes de análise sejam visualizados no contexto da sequência, em seus seis *frames* de leitura. Especialmente útil para genomas compactos de bactérias, arqueia ou eucariotos menores, pode ser utilizado para visualização de pequenos genes ou genomas completos. Sendo um programa específico para visualização e anotação de dados genômicos, não realiza tarefas como predição de genes ou busca por ORFs (RUTHERFORD et al., 2000; SANGER, 2014).

Para a visualização de dados, o Artemis desenha um gráfico da sequência levando em consideração o tamanho do genoma, tamanho e posição de cada *feature*, possibilitando a utilização de *zoom* para análise de regiões específicas. Também são exibidas propriedades da sequência como conteúdo G+C, proporção G/C*,* e conteúdo G+C específico de um frame. As informações exibidas são extraídas de arquivos no formato EMBL e GenBank. A interface do programa pode ser visualizada na Figura 3.

FIGURA 3 - INTERFACE DO PROGRAMA ARTEMIS



FONTE: adaptado de RUTHERFORD et al., 2000

É possível observar na imagem, em A1, o gráfico de conteúdo G+C de um *frame* em específico. Em A2 estão os seis *frames* de leitura, nos quais as barras verticais pretas representam códons de término e as barras horizontais coloridas as *features*. Em A3 estão os mesmos *frames* exibidos a nível de nucleotídeo, tendo logo abaixo a descrição de cada *feature*, com sua respectiva posição e descrição. Em B está um panorama geral da sequência, exibindo dados estatísticos como número de bases, densidade, tamanho médio dos genes, dentre outros. Em C é possível observar um gráfico de propriedades de uma região codificante. Em D está a caixa para edição dos *qualifiers*.

Com relação à anotação, o programa possibilita que *features* presentes nas sequências sejam anotadas ou modificadas. É possível atribuir diversas propriedades pré-definidas ou customizadas para cada elemento. Os dados anotados podem ser salvos no próprio arquivo ou em arquivos diferentes, conforme desejo do usuário. Também é possível enviar as *features* para o programa BLAST e visualizar os dados na própria interface do Artemis.

O Artemis está disponível para os sistemas operacionais Linux, MacOS e Windows, desde que estes possuam o Java instalado. Também é possível

executar o programa através de um *Web applet*, ou seja, utilizar o programa diretamente de um servidor como se ele estivesse instalado na máquina do usuário.

Não é possível integrar o Artemis a outras ferramentas, conforme (RUTHERFORD et al., 2000), dados originados por softwares externos podem ser visualizados e anotados no programa desde que estejam nos formatos de arquivo aceitos.

2.3 APLICAÇÕES SIMILARES AO WEB GAAT

Esta sessão tem como objetivo analisar programas com propósitos similares ao Web GAAT afim de identificar pontos positivos e negativos da ferramenta frente a outros recursos disponíveis.

2.3.1 GAAT

GAAT *(Genome Assembly and Analysis Tool)* é uma ferramenta de anotação automática de genomas desenvolvida no Laboratório de Bioinformática do Núcleo da Fixação Biológica de Nitrogênio da UFPR para anotação e revisão de genomas (TIEPPO, 2011).

Embora seja uma aplicação Web, é necessário instalar o GAAT em cada computador em que o programa será utilizado. Tal necessidade faz com que cada máquina seja também um servidor Web, característica que requer configurações de hardware mais altas em relação a computadores convencionais, além de um conhecimento técnico bastante específico para configuração e disponibilização.

Pisa (2008) destaca as características do programa em sua última atualização:

- Identificação automática de ORFs obtidas a partir do envio de sequências no formato FASTA utilizando o programa Glimmer.
- Identificação de sequências de ligação de ribossomo (RBS) em cada ORF, utilizando o programa RBSfinder.
- Identificação de sequências de RNA transportadores, utilizando o programa tRNAscanSE.

- Busca por similaridade nas sequências contidas na base de dados do NCBI através do programa BLAST.

As funcionalidades descritas acima são possíveis pela integração dos programas mencionados por meio de scripts que são executados automaticamente no servidor a partir de ações realizadas na interface de usuário.

Em comparação com o GAAT nota-se bastante semelhança com os objetivos propostos neste trabalho, pois, é também objetivo desenvolver uma ferramenta com potencial para substituir o programa.

Questões como a necessidade de instalação do GAAT em cada computador onde será utilizado bem com a impossibilidade de divisão de atividades são tratadas pelo Web GAAT, facilitando sua utilização.

Outra possibilidade identificada ao analisar o programa GAAT foi a de adaptação de seus scripts para tratamento de arquivos FASTA e integração com os programas Glimmer e tRNAscanSE para atender às necessidades e objetivos do Web GAAT.

2.3.2 Web Apollo

Integrante do acervo de aplicações disponibilizadas pelo GMOD *(Generic Model Organism Database)*, o Web Apollo é uma ferramenta web para visualização, edição e anotação de genes, a primeira aplicação deste tipo a possibilitar anotações colaborativas em tempo real (Lee et al, 2013).

Desenhada e distribuída com foco em trabalho comunitário, onde pesquisadores podem estar distribuídos em regiões diferentes (GMOD, 2016), o programa possui em sua gama de recursos:

- Histórico com rastreamento dos dados anotados, compreendendo navegação por versões anteriores de uma anotação, possibilidade de edição e opções para desfazer ou refazer dados alterados.
- Atualização em tempo real, as edições em um cliente (computador) são disponibilizadas automaticamente para outros clientes.
- Gerenciamento de usuários, compreendendo autenticação e permissões de acesso e edição das sequências.

- Processo de curadoria em dois estágios, anotações podem ser realizadas em uma área de trabalho temporária e posteriormente publicadas em um banco de dados.
- Carregamento de dados diretamente a partir de arquivos GFF3, BigWig e BAM armazenados em um servidor ou no computador do usuário.
- Exportação dos dados anotados para arquivos GFF3 ou FASTA.

Assim como proposto pelo Web GAAT, o Web Apollo possui arquitetura modularizada. As interações do usuário são mediadas pelo JBrowse (SKINNER et al., 2009), um software Web para visualização de genomas, também integrante do GMOD. Todas as interações que ocorrem na interface do usuário (UI) são capturadas e tratadas pelo Web Apollo, o qual utiliza módulos de anotação hospedados no servidor para dinamicamente processar os dados recebidos e devolver para o usuário o resultado destas ações.

Em comparação com o Web GAAT, nota-se grande semelhança entre recursos propostos por este trabalho e os recursos existentes no Web Apollo em três pontos:

1. Gerenciamento de usuários, provendo autenticação no sistema e autorização de acesso à dados para anotação.
2. Possibilidade de visualização de versões anteriores de uma anotação para uma possível reversão de dados anotados.
3. Possibilidade de atividade colaborativa.

O Web Apollo possui como diferenciais o tratamento de arquivos GF#, BigWig e BAM, a atualização em tempo real e o processo de curadoria em dois estágios. Já o Web GAAT possui como diferenciais a possibilidade de identificação automática de ORFs, tRNAs e predição de genes, a conversão automática de sequências de nucleotídeos em sequência de proteínas, a possibilidade de envio de sequências para BLASTN e BLASTP. Tais características posicionam a ferramenta proposta como uma alternativa considerável tendo em vista as facilidades oferecidas pela automação das tarefas mencionadas.

2.4 BANCO DE DADOS BIOLÓGICOS

Bancos de dados biológicos são coleções organizadas de informações sobre organismos, recolhidas a partir de experiências científicas, literatura publicada, tecnologia experimental e análise computacional (ATTWOOD et al, 2011).

Para Baxevanis (2009), estes bancos desempenham um papel central na bioinformática, oferecendo aos cientistas a oportunidade de acessar uma grande variedade de dados biologicamente relevantes, incluindo as sequências genômicas de uma gama cada vez mais ampla de organismos.

Andreatta (2013) relata que é possível considerar seguintes bancos de dados biológicos como os principais:

- GenBank: Mantido pelo NCBI (*National Center for Biotechnology Information)*, é um banco de dados público de sequências de nucleotídeos e anotação para apoio bibliográfico e biológico;
- EMBL-ENA: O *European Nucleotide Archive* (ENA), pertence ao *European Molecular Biology Laboratory* (EMBL) é o banco europeu de sequências de nucleotídeos;
- DDBJ: Mantido pelo *National Institute of Genetics* (NIG), o *DNA Data Bank of Japan* (DDBJ) é o banco de sequências de nucleotídeos da Ásia.

Estes três bancos formam o INSDC, *International Nucleotide Sequence Database Collaboration* e possuem rotinas diárias para sincronização de seus acervos dividindo as informações a mantendo-se atualizados. (ANDREATTA, 2013).

2.5 BIO* TOOLKITS

Durante a etapa de pesquisa de tecnologias para desenvolvimento do Web GAAT, foi possível encontrar diversos recursos úteis para a criação de aplicações relacionadas a anotação de genomas. Chamados de Bio* Toolkits (MANGALAN et al, 2002), estes recursos reúnem funcionalidades essenciais, como:

- Interpretadores (*parsers*) de arquivos Fasta e GenBank, muito úteis para manipular os dados presentes nestes arquivos;
- Comparadores de sequência, recurso importante para identificar similaridade entre organismos;
- Dicionários, úteis para converter sequências de nucleotídeos em aminoácidos;
- Acesso a sequências de nucleotídeos a partir de bases de dados locais e remotas.

Segundo (MANGALAN, 2002), os principais Bio* Toolkits disponíveis são:

- BioPerl: É o pioneiro, escrito em Perl, foi utilizado no projeto de sequenciamento do genoma humano. É um projeto *open source* apoiado pela Open Bioinformatics Foundation;
- BioPython: Também é um projeto *open source* e apoiado pela Open Bioinformatics Foundation;
- BioJava: Assim como BioPerl e BioPython, é um projeto *open source*, mas escrito em Java, linguagem de programação proprietária da empresa Oracle.

Ainda segundo (MANGALAN, 2002), os três Bio*Toolkits apresentam as mesmas possibilidades de utilização. Desta forma, a escolha por um deles deve ser mediada por outros fatores englobando o desenvolvimento de um projeto de software, como conhecimento dos envolvidos sobre as linguagens de programação, infraestrutura e possibilidade de integração com outros recursos.

Para o Web GAAT, devido ao ambiente de produção onde a plataforma foi disponibilizada, os Bio*Toolkits BioPerl e BioPython foram configurados para uso.

2.6 APLICAÇÕES WEB

Segundo NATIONS (2016) em computação, aplicação Web designa, de forma geral, sistemas de informática projetados para utilização a partir de um navegador (*browser*) ou aplicativos desenvolvidos utilizando tecnologias Web.

Nos tópicos a seguir são apresentados os elementos envolvidos no processo de desenvolvimento de softwares para Web, bem como os conceitos que envolvem a criação destas aplicações. Inicialmente apresenta-se a estrutura

de uma aplicação Web e o modelo MVC, amplamente utilizado, em seguida são apresentadas tecnologias e metodologias relacionadas ao desenvolvimento do trabalho.

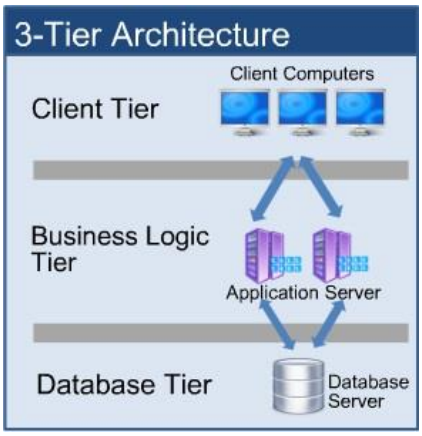
2.6.1 Estrutura de Aplicações Web

Em termos estruturais, as aplicações Web são geralmente divididas em blocos lógicos chamados *tiers*, sendo que para cada camada é atribuído um papel específico (PETERSEN, 2008). Ainda de acordo com (PETERSEN, 2009), a estrutura mais comum é a de três camadas de aplicação:

- Camada de apresentação (*client tier*);
- Camada de aplicação (*business logic tier*);
- Camada de armazenamento (*database tier*).

Seguindo essa abordagem é possível concluir que o navegador é a primeira camada (apresentação), na qual os dados são apresentados para interação entre usuário e sistema. Na camada de aplicação, segunda camada, o sistema realiza o controle da aplicação utilizando alguma linguagem de programação. Por fim, o terceiro nível armazena os dados relativos ao uso do sistema. O fluxo de comunicação entre as três camadas pode ser observado na Figura 4.

FIGURA 4 - FLUXO DE COMUNICAÇÃO ENTRE AS TRÊS CAMADAS DE APLICAÇÃO



FONTE: THATAVARTHI; SURESH, 2013.

Conforme observado na Figura 4, o modelo MVC é distribuído em três níveis ou camadas: *Database Tier* (armazenamento), *Business Logic Tier*

(aplicação) e *Client Tier* (apresentação). Na camada de apresentação são realizadas as interações do usuário com o sistema. Estas interações são tratadas pela camada de aplicação que, por sua vez, se comunica com a camada de armazenamento e devolve para o nível de apresentação as informações relativas ao uso do sistema.

A divisão estrutural de uma aplicação Web em três camadas assemelha-se ao padrão de projeto *Model View Controller*, comumente chamado apenas de MVC.

Tanto a definição de padrões de projeto quanto a definição de MVC são abordados a seguir.

2.6.2 Padrões de projeto

Padrões de projeto, do inglês *design patterns*, é uma solução geral para um problema que ocorre com frequência dentro de um determinado contexto no projeto de um software (GAMMA et al, 1995). Um padrão de projeto não é um projeto finalizado, que pode ser transformado automaticamente em um software com uma funcionalidade específica, ele é uma descrição, um guia de melhores práticas formalizadas para o desenvolvimento de uma aplicação.

Ainda segundo (GAMMA et al, 1995), um padrão possui quatro elementos essenciais:

- O nome, que pode ser usado para descrever um padrão de projeto;
- Um problema ou cenário, que descreve quando aplicar o padrão;
- A solução, que informa os elementos que compõe o padrão, quando aplicá-los, seus relacionamentos, responsabilidades e colaborações;
- As consequências, que são resultados e vantagens e desvantagens de se aplicar um determinado padrão.

Os padrões de projeto visam facilitar a reutilização de soluções comprovadas na fase de planejamento de um determinado software, estabelecendo um vocabulário comum para desenho, facilitando comunicação e documentação dos projetos. (GAMMA et al, 1995).

Em projetos de softwares para Web, o padrão MVC é o mais utilizado por ser semelhante à estrutura de aplicações para Internet.

2.6.2.1 MVC

Apesar de ter sido desenvolvido originalmente para computação pessoal, o MVC foi amplamente adaptado como uma arquitetura para aplicações Web (LEFF; RAYFIELD, 2001).

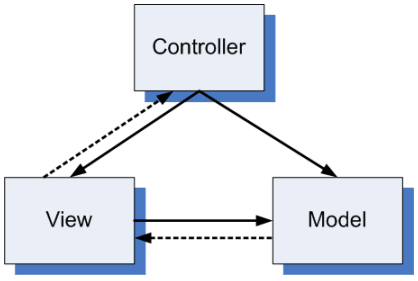
Conforme (GAMMA et al, 1995), o MVC é um padrão de projeto, composto por três camadas ou objetos: *Model*, *View* e *Controller*, os quais permitem dividir as funcionalidades do sistema em diferentes camadas com responsabilidades distintas para aumentar a flexibilidade, manutenção e reutilização daquilo que foi produzido.

As responsabilidades de cada camada estão descritas a seguir:

a) *Model*: Responsável por representar as informações presentes no software. Geralmente utiliza banco de dados para armazenamento e consulta de informações;
b) *View*: Responsável por apresentar as informações aos usuários finais, geralmente por meio de navegadores Web;
c) *Controller*: Responsável por controlar o fluxo das informações do sistema conforme dados exibidos e interações do usuário.

Na Figura 5 podemos observar o fluxo de comunicação entre as camadas. Nota-se a semelhança do padrão MVC com a estrutura de desenvolvimento de softwares para Web. Tanto o modelo proposto pelo padrão de projeto, quanto a definição estrutural, dividem uma aplicação em três camadas: uma camada responsável por apresentar as informações ao usuário, uma camada responsável por armazenar essas informações e outra responsável por controlar o fluxo de interação entre as partes.

FIGURA 5 - PADRÃO DE PROJETO MVC



FONTE: HARVARD, 2010.

A divisão de uma aplicação Web em camadas é bastante útil em situações nas quais equipes diferentes trabalham no desenvolvimento do software, pois, possibilita que cada camada seja desenvolvida separadamente. Também é bastante útil quando deseja-se atualizar apenas partes da aplicação. O modelo MVC permite que as camadas recebam atualizações separadamente, desde que não impactem nas demais (LEFF; RAYFIELD, 2001).

2.6.2.2 Banco de Dados

Tanto na estrutura de desenvolvimento de softwares para Web como no padrão de projeto MVC, uma camada é responsável pelo armazenamento de informações. Este armazenamento pode ser feito de várias maneiras, na memória do computador, em arquivos e também em programas específicos para essa finalidade, como os bancos de dados.

Silberschatz et al (2006) definem banco de dados como coleções organizadas de dados inter-relacionados comumente operados por um sistema de gerenciamento de banco de dados (SGBD) a fim de fornecer métodos para recuperar informações de maneira tanto conveniente quanto eficiente.

É possível exemplificar várias utilizações para um banco de dados. Em sites de compra, bancos de dados são usados para armazenar informações sobre mercadorias, como nome, fabricante, preço e disponibilidade em estoque. Na área da biologia é possível utilizar os bancos de dados biológicos para consultas sobre diversos genomas. Sites como NCBI e Uniprot possuem vasta coleção de dados sobre os mais variados genes de diferentes organismos.

Para desenvolvimento do Web GAAT, foi essencial a utilização de métodos eficientes de recuperação das informações armazenadas por meio do banco de dados, pois, um dos objetivos é extrair e armazenar dados sobre sequências para que essas possam ser visualizadas e anotadas. Segundo (MIRANDA, 2016), os principais SGDBs de mercado são:

1. **ORACLE:** SGDB líder de mercado. Atualmente na versão 12c, possui três distribuições: Enterprise Edition, Standard Edition e Standard Edition One. (ORACLE, 2016). É um SGBD pago, em sistema de registro único.

2. **MySQL:** É um SGBD que tem como foco sistemas online, também pertence a ORACLE. Lançado em 1996 está na versão 5, apresentada em fevereiro de 2016. O seu grande diferencial é ser um sistema *Open Source*, isto é, não é necessário adquirir uma licença para utilização.
3. **SQL Server:** Embora produto da Microsoft, não necessariamente precisar ser utilizado em conjunto com outras soluções da empresa. (MICROSOFT SQL SERVER, 2016). É um SGBD amplamente utilizado, ficando atrás apenas do ORACLE e MySQL. Para utilização do SQL Server é necessário comprar o produto de acordo com um dos modelos de licença oferecidos pela Microsoft.
4. **PostgreSQL:** Também *Open Source* é muito utilizado para sistemas web, principalmente em pequenos sistemas (MIRANDA, 2016).
5. **MongoDB:** Banco de dados *Open Source*, idealizado para armazenamento de dados em arquivos (Mongo DB, 2016).

Com exceção do MongoDB, os demais SGBDs apresentam basicamente os mesmos recursos de segurança, armazenamento e recuperação de dados. A escolha por um deles deve ser mediada por fatores como tamanho da aplicação a ser desenvolvida, custo, recursos de infraestrutura e tecnologias envolvidas em sua utilização. Para o Web GAAT foi escolhido o banco MySQL, por possuir licença gratuita e ser robusto. Este será especificado no Capítulo 3, sobre materiais e métodos.

2.6.2.3 Frameworks e bibliotecas

Segundo a definição de Riehle (2000), em computação, um framework corresponde a um modelo abstrato de software que provê aos usuários um conjunto de funcionalidades prontas para uso, mas que podem ser alteradas de acordo com a necessidade específica de cada programa.

Implementados em uma linguagem de programação específica, se assemelham aos padrões de projeto no intuito de oferecer uma série de soluções que podem ser replicadas, mas diferenciam-se ao disponibilizar código pronto para uso.

Em aplicações Web, muitos frameworks seguem o padrão MVC para separar o modelo de dados com as regras de negócio da interface do usuário, o

que é considerado uma boa prática, uma vez que modulariza a aplicação, promove reutilização de código e permite que várias interfaces sejam aplicadas (RUSSEL; COHN, 2013).

A utilização de um framework está diretamente ligada à tecnologia em que um sistema é desenvolvido. Podendo existir mais de um para diferentes tecnologias, a escolha deve ser também mediada pela familiaridade dos envolvidos com o projeto, curva de aprendizado e, principalmente, finalidade do projeto, para determinar se a escolha possui o conjunto de funcionalidades necessário para obtenção dos objetivos traçados.

Com relação a bibliotecas, estas se assemelham muito aos frameworks por disponibilizarem um conjunto de funcionalidades prontas para uso, mas diferenciam-se ao serem projetadas para atender a porções específicas de uma aplicação, como interpretação de determinados tipos de arquivo, desenho de gráficos ou controlar o acesso de usuários em um sistema, fornecendo mecanismos para cadastro, login e senha. Bibliotecas podem ser utilizadas em conjunto com frameworks para atender apenas determinadas funções de uma aplicação (RIEHLE, 2000).

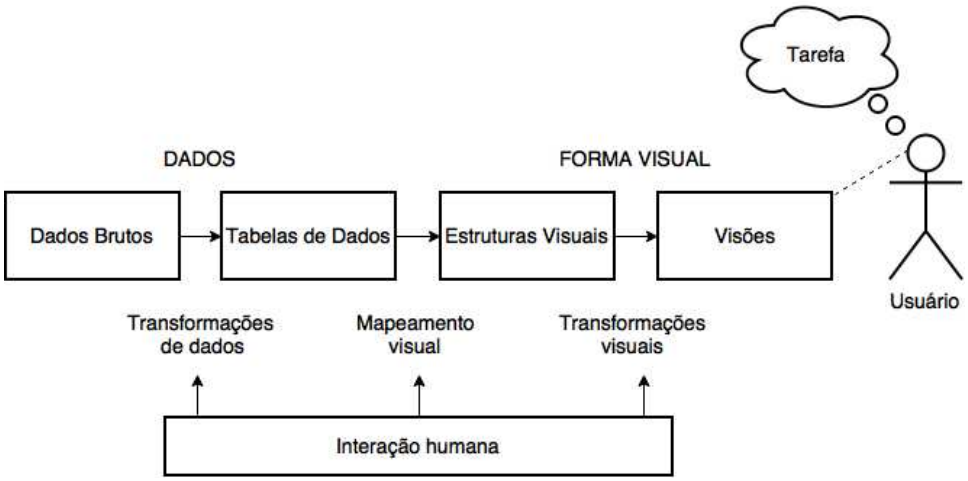
2.7 VISUALIZAÇÃO DA INFORMAÇÃO

A Visualização da Informação (VI) tem como objetivo potencializar compreensão da informação pelo usuário por meio de recursos gráficos interativos (FREITAS et al., 2001). Atendendo a um paradigma de escrita e linguagem, a VI está geralmente relacionada a um contexto histórico, onde imagens em uma sequência ou ordem expressam alguma informação e transmite uma mensagem (DIAS; CARVALHO, 2007). O contexto histórico, mencionado pelo autor, condiz com a definição de (CARD et al., 1999), onde as imagens devem ser geradas com base no relacionamento de dados e na experiência do usuário de apresentar dados desse tipo.

Nascimento e Ferreira (2011) relatam a VI é muito comum em infográficos exibidos frequentemente em mídias digitais e impressas, pois sua utilização pode ampliar a cognição sobre dados abstratos quando comparada a outros formatos para transmissão e análise de dados.

Nascimento e Ferreira (2011) descrevem o modelo composto por três etapas para a Visualização da Informação (CARD et al., 1999), que pode ser observado na Figura 6.

FIGURA 6 - MODELO PARA VISUALIZAÇÃO DA INFORMAÇÃO



FONTE: adaptado de CARD et al., 1999, DIAS; CARVALHO, 2007 e NASCIMENTO; FERREIRA, 2011.

A primeira etapa, Transformação de Dados, consiste na obtenção das informações que devem ser apresentadas. Nesta fase dados brutos são tratados e reorganizados em um modelo estruturado, geralmente em uma tabela, onde as linhas representam os elementos a serem exibidos e as colunas suas propriedades. O tratamento de dados que ocorre nesta fase da VI pode compreender cálculos estatísticos, eliminação de redundâncias ou filtragem das informações relevantes.

Na segunda etapa, Mapeamento Visual, os dados tratados são organizados em uma estrutura visual, podendo ser composta por três partes: substrato ou meio espacial, marcas visuais e propriedades gráficas.

O substrato espacial se caracteriza pelo espaço para visualização, onde os dados podem ser apresentados em um dos modelos abaixo:

- Eixo não estruturado, onde a posição espacial não interfere nas informações;
- Eixo nominal, que consiste na divisão de sub-regiões para agrupamento dos dados de acordo com alguma característica;
- Eixo ordenado, onde a posição de um elemento tem importância.

- Eixo quantitativo, no qual a região ocupada por um elemento representa uma métrica.

As marcas visuais representam os itens dos dados, geralmente através de formas geométricas ou figuras (ícones) que representam graficamente os elementos.

Por fim, são atribuídas as propriedades gráficas que dão características às marcas visuais. Cor, tamanho, volume e posição são exemplos de atributos que podem ser utilizados para potencializar as características de cada item.

A terceira etapa do modelo, Transformações Visuais, consiste na aplicação de interatividade aos itens de dados. O objetivo é poder explorar os elementos com mais profundidade, revelando propriedades ou relações invisíveis em um panorama geral. Nesta etapa os recursos computacionais são essenciais para possíveis cálculos, alterações de cenário ou transformações nas marcas gráficas e suas propriedades. Para Lima (2011), os principais recursos de computação gráfica utilizados nesta etapa são:

- Aproximação ou *zoom,* o qual possibilita a visualização aproximada ou em maior escala de uma região de interesse;
- Transformação, que se caracteriza pela alteração de cor ou forma dos elementos;
- Revelação de detalhes, que consiste na exibição de maiores informações sobre um item explorado.

Lima (2011) ainda define que uma boa visualização parte de uma visão geral simplificada, contendo apenas dados espalhados ou agrupados e suas propriedades gráficas essenciais, para uma visão detalhada, interativa e gradual, guiada pelo interesse do usuário.

3 MATERIAIS E MÉTODOS

Este capítulo apresenta os instrumentos aplicados no desenvolvimento deste projeto. Inclui-se também o detalhamento da arquitetura proposta, bem como as tecnologias envolvidas, especificação e desenvolvimento da plataforma.

3.1 Visão geral

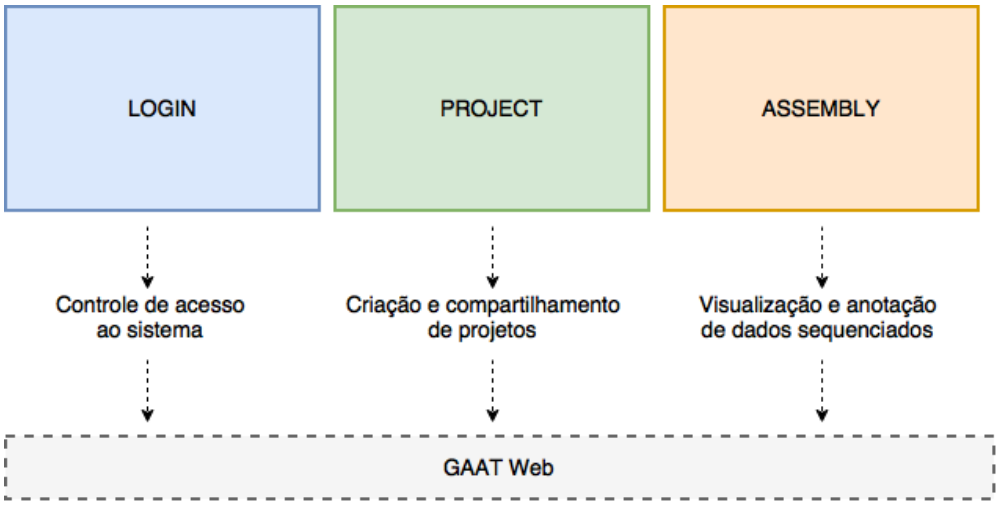
O presente trabalho descreve a construção de um software Web para visualização e anotação de genomas, nomeado Web GAAT. A plataforma objetiva-se a reunir em uma interface única recursos utilizados no processo de análise e anotação de dados sequenciados, para pesquisadores que desejam trabalhar em projetos individuais ou compartilhados. Além disto, o projeto foi especificado e construído de forma flexível, possibilitando a adição de novas funcionalidades e a expansão de seus recursos.

De acordo com os objetivos propostos pela plataforma, os recursos especificados e desenvolvidos para o Web GAAT abrangem as seguintes funcionalidades:

a) Identificação automática de ORFs, tRNAs e predição de genes;
b) Possibilidade de realizar anotação sobre dados obtidos;
c) Conversão automática de sequências de nucleotídeos em sequência de proteínas;
d) Possibilidade de envio de sequências para Web BLASTN e Web BLASTP;
e) Visualização de genomas parciais e completos;
f) Possiblidade de criação de projetos individuais;
g) Possibilidade de compartilhamento de projetos para que mais de um pesquisador tenha acesso aos dados;
h) Armazenamento de histórico sobre dados anotados;
i) Arquitetura flexível para adição de novas funcionalidades;
j) Código de fácil atualização e manutenção.
k) Substituição do programa homônimo GAAT.

Desenvolvido utilizando o padrão de projetos MVC, o conjunto de funcionalidades proposto pela plataforma foi dividida em três módulos, conforme Figura 7-1, na qual é possível observar que cada módulo é responsável por agrupar um conjunto específico de funcionalidades: Login, que gerencia o fluxo de usuários no sistema; Project, responsável por gerenciar projetos de anotação de genomas e compartilhamento dos mesmos e, por fim, Assembly, que reúne todas as funcionalidades relacionadas à visualização e anotação de dados sequenciados.

FIGURA 7 - DIVISÃO MODULAR DA APLICAÇÃO



FONTE: O autor (2016).

A divisão modular da aplicação tem como objetivo aumentar a flexibilidade da plataforma. Ao agrupar um conjunto de atividades em um módulo específico, rotinas de manutenção podem ser realizadas de forma isolada. Além disto, a abordagem permite que novos módulos, com novas funcionalidades, sejam criados e adicionados à plataforma, aproveitando recursos já implementados e possibilitando a utilização em uma mesma interface.

Cada módulo do Web GAAT, responsável por um conjunto específico de funcionalidades, reuniu diferentes tecnologias em seu desenvolvimento. Nos módulos Login e Project, funcionalidades disponibilizadas pelo framework CodeIgniter (CODEIGNITER, 2017) possibilitam que usuários gerenciem seus projetos e o acesso a eles, informando quem pode acessa-los. No módulo Assembly, para as atividades relacionadas à identificação automática de ORFs, um script em Perl, baseado no script utilizado pelo programa GAAT, procura nos

contigs presentes nos arquivos Fasta trechos que começam com um códon de início específico. Para esta finalidade também foi integrado ao Web GAAT o software GLIMMER; outro script em Perl, também similar ao script presente no GAAT, é responsável por interpretar os dados gerados pelo programa e gravar no banco os resultados. Para a busca de tRNAs, outro script em Perl, também similar ao script utilizado pelo programa legado, analisa os resultados do tRNAscan-SE, que também foi integrado ao Web GAAT. Por fim, para a visualização e anotação de dados, uma interface interativa, baseada em princípios de VI e construída utilizando a biblioteca BioCircos (OMICTOOLS, 2017), possibilita a interação do usuário com os dados armazenados no programa.

Nos tópicos a seguir estão relatados com maior especificidade as tecnologias e metodologias utilizadas para a construção da plataforma, de seus módulos e funcionalidades, bem como os recursos de infraestrutura utilizados pelo sistema.

3.2 AMBIENTE DE PRODUÇÃO

Para o ambiente de produção, local onde o Web GAAT foi disponibilizado, os seguintes recursos foram configurados em um servidor disponibilizado pela UFPR para implementação da plataforma:

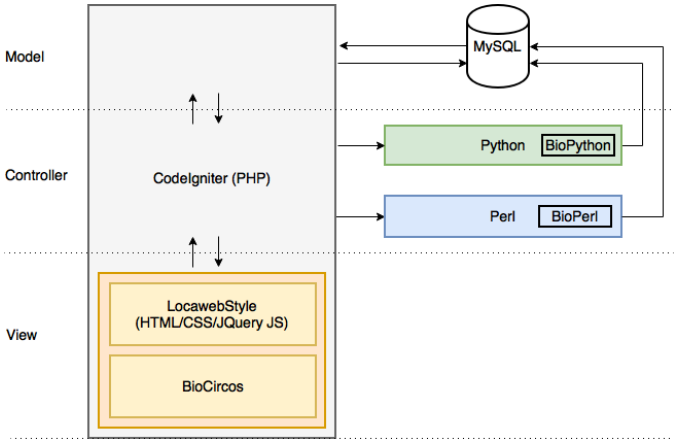
- Sistema operacional Ubuntu versão 15.10
- Servidor Web Apache versão 2.4.12
- Banco de dados MySQL versão 14.14 distribuição 5.6.30
- PHP versão 5.6.11
- Perl 5 versão 5.20.2
- Python versão 2.7

A escolha por estes recursos foi mediada pela análise das funcionalidades que o programa se objetiva a disponibilizar, e também pelo fato de serem tecnologias *open-source*, amplamente utilizadas e de distribuição livre, permitindo que possam ser atualizadas sem necessidade de aquisição de novas licenças.

3.3 ARQUITETURA

No intuito de construir uma aplicação flexível, o software foi criado utilizando o padrão de projetos MVC. Esta abordagem permite que camadas específicas do programa sofram modificações, facilitando sua manutenção e futuros desenvolvimentos. A Figura 8 ilustra o padrão arquitetural do Web GAAT dividido em três camadas e indicando também a utilização dos frameworks CodeIgniter e LocawebStyle, e da biblioteca BioCircos JS.

FIGURA 8 - ARQUITETURA PROPOSTA PARA O WEB GAAT



FONTE: O autor (2016).

Conforme visualizado na Figura 8, o framework CodeIgniter abrange toda a aplicação, gerenciando as três camadas do modelo MVC: *View* (nível mais abaixo, *Controller* (nível intermediário) e *Model* (nível mais acima).

Na camada *View* ocorrem todas as interações do usuário com o Web GAAT. A camada *Model* destina-se a armazenar os dados relativos ao uso do sistema, como o armazenamento de dados sobre sequências e recuperação dos mesmos. Por fim, a camada intermediária *Controller* gerencia todo o fluxo entre as interações do usuário com a View e os dados armazenados no *Model*. Também é possível observar na figura que os scripts escritos em Perl e Python, relacionados ao módulo Assembly, encontram-se no nível da camada *Controller*, porém estes enviam os dados diretamente para o banco, não utilizando a camada *Model* para as interações.

Na camada *View*, por sua vez, é possível observar a presença do framework *Open Source* LocawebStyle (LOCAWEB, 2017). A utilização do

framework objetiva-se a estabelecer um padrão de código específico para esta camada. O LocawebStyle possui uma biblioteca com diversos elementos prontos para uso, definindo assim um padrão gráfico para o sistema, desta forma cumprindo o objetivo de fornecer uma interface única para interações do usuário com a plataforma.

Conforme relatado na Visão Geral (Seção 3.1), a fim de aumentar a flexibilidade da plataforma, os objetivos específicos do Web GAAT foram divididos e organizados em grupos de acordo com seu papel funcional no sistema. Para cada grupo um módulo foi desenvolvido com a finalidade de fornecer um conjunto completo de funcionalidades. A seguir estão listados os módulos criados e a que atividades são destinados.

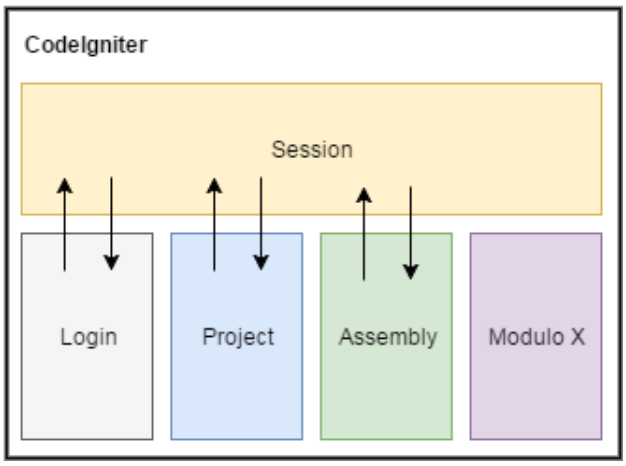
- Login: Módulo que possibilita a criação de usuários e provê acesso autenticado ao Web GAAT. Este módulo foi criado com o objetivo aumentar a segurança da plataforma, permitindo exclusividade nas interações de um usuário com o sistema.
- Project: Módulo destinado à criação e compartilhamento de projetos entre usuários do sistema. O módulo possibilita que um usuário crie um projeto individual ou compartilhe este com outros usuários, possibilitando assim o trabalho colaborativo.
- Assembly: Módulo que reúne as funcionalidades relacionadas à visualização e anotação de genomas. Fazem parte deste módulo a predição de genes, a busca automática por ORFs e tRNAs, a visualização e anotação e armazenamento de histórico sobre dados anotados.

A decisão de agrupar atividades em partes funcionais flexibiliza a arquitetura da aplicação, além de possibilitar também a reutilização de recursos já existentes (TECHOPEDIA, 2018). Por exemplo, um novo módulo poderia utilizar somente Login para prover seu acesso, ou poderia utilizar Project para criação de projetos individuais ou compartilhados. Outra característica da divisão modular é a facilidade de manutenção, que pode ocorrer apenas em módulos isolados, sem afetar funcionalidades relativas a outras partes (módulos) da plataforma.

Os módulos listados e a comunicação entre eles podem ser observados na Figura 9. Note a divisão modular da plataforma. A comunicação entre os

módulos é realizada a partir do uso da sessão *(session).* Dados sobre qual usuário está autenticado no sistema e em qual projeto o mesmo está trabalhando ficam assim disponíveis para todos os módulos. Ainda nesta figura, o Módulo X ilustra o desenvolvimento de um novo módulo, podendo este seguir o mesmo padrão de comunicação dos demais ou ser totalmente independente.

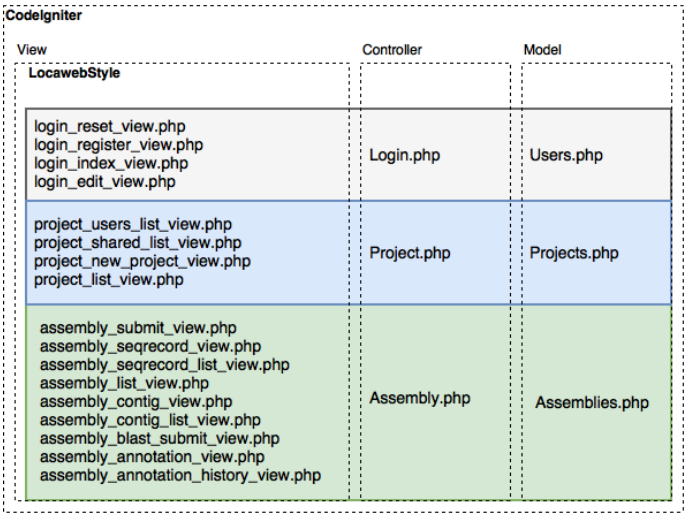
FIGURA 9 – COMUNICAÇÃO ENTRE OS MÓDULOS DA APLICAÇÃO



FONTE: O autor (2016).

Com relação à utilização do CodeIgniter, no aspecto de uma aplicação modular, justifica-se a utilização do framework pois este possibilita um padrão de desenvolvimento para os módulos. A metodologia proposta pelo CodeIgniter auxilia na divisão de uma aplicação em partes funcionais, além de estabelecer também um formato de comunicação seus elementos. Na Figura 10 é possível visualizar a arquitetura da aplicação de acordo com o framework CodeIgniter. É possível visualizar também que cada módulo possui um conjunto específico de arquivos. Por fim, nota-se que o framework LocawebStyle é utilizado pela View de todos os módulos, estabelecendo um padrão de interface para a plataforma.

FIGURA 10 - ARQUITERTURA DA APLICAÇÃO DE ACORDO COM O FRAMEWORK CODEIGNITER



FONTE: O autor (2016).

No desenvolvimento do Web GAAT foram utilizadas as seguintes versões dos frameworks CodeIgniter (https://codeigniter.com) e LocawebStyle (http://opensource.locaweb.com.br/locawebstyle/):

- CodeIgniter versão 3.0.3
- LocawebStyle versão 3.9.0

Com relação ao BioCircos JS, conforme visualizado na Figura 8, foi utilizada a versão 1.1.1 da biblioteca. A utilização da biblioteca será justificada no item 3.4.3 deste capítulo.

3.4 DESENVOLVIMENTO

Conforme descrito na sessão Arquitetura, o Web GAAT foi divido em módulos que agrupam um conjunto específico de funcionalidades a fim de atender aos objetivos específicos do programa. Nós tópicos a seguir estes módulos são descritos com maior especificidade.

3.4.1 Login

O módulo login destina-se a gerenciar a autenticação de usuários no sistema, para que estes possam acessar o Web GAAT e utilizar seus recursos. Este módulo compreende as seguintes funcionalidades:

- Login e logout do sistema;

- Registro de novos usuários;
- Alteração de dados do perfil do usuário;
- Recuperação de senha.

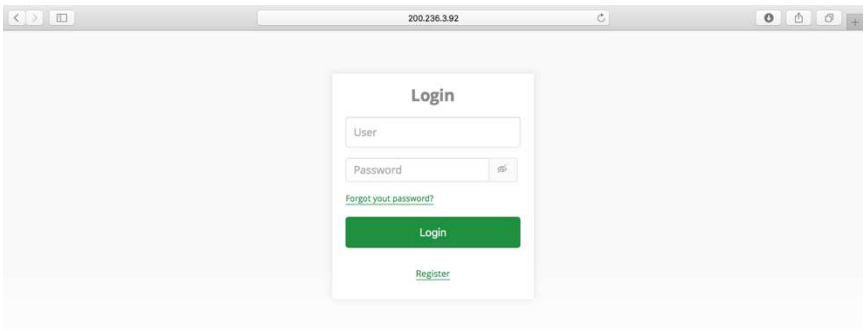
Para registrar-se no Web GAAT, o usuário deve informar um conjunto de dados que serão validados pelo sistema. São eles:

- Name: Campo onde o usuário deve informar seu nome. O preenchimento é obrigatório e o valor informado deve conter apenas letras e no máximo 50 caracteres;
- Last name: Campo onde o usuário deve informar seu sobrenome; Campo também obrigatório, e o valor informado deve conter apenas letras e no máximo 50 caracteres;
- E-mail: Campo onde o usuário deve informar um e-mail que servirá como login. Deve ser um e-mail com um formato válido, exemplo: nome@dominio.com e único no banco de dados do sistema
- Confirm e-mail: Campo para confirmação do valor informado no campo E-mail. Destina-se a ajudar o usuário a identificar possíveis erros de digitação neste campo;
- Password: Campo para criação de uma senha. Deve conter de 4 a 16 caracteres alfanuméricos, podendo conter letras de A à Z, maiúsculas e minúsculas e números reais, de 0 a 9.
- Confirm password: Campo para confirmação do valor digitado em Password. Destina-se a ajudar o usuário a identificar possíveis erros de digitação neste campo.

Caso o valor informado pelo usuário não esteja de acordo com as regras estabelecidas, o sistema notificará quais dados devem ser corrigidos e o motivo pelo qual a informação não foi aceita.

A tela de login, com os campos para acesso ao Web GAAT e as opções para registro e recuperação de senha pode ser observada na Figura 11.

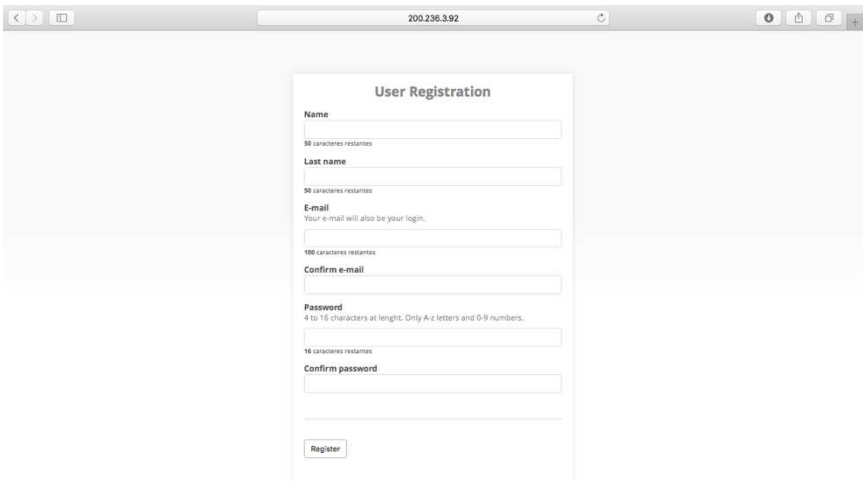
FIGURA 11 - TELA DE LOGIN



FONTE: O autor (2016).

A tela referente ao registro de usuários, *User registration*, pode ser visualiza na Figura 12.

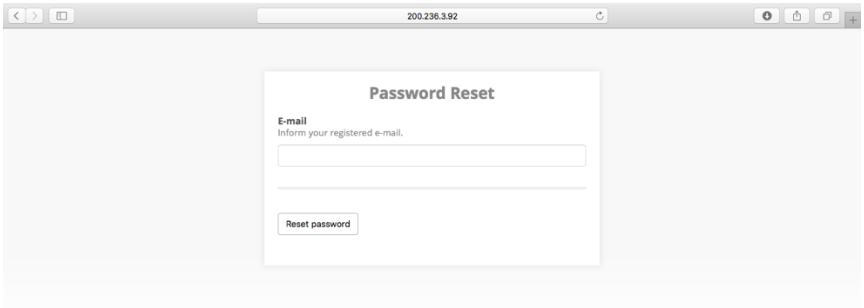
FIGURA 12 - TELA DE REGISTRO DE USUÁRIOS



FONTE: O autor (2016).

Em caso de esquecimento de senha, é possível solicitar o envio de uma nova senha para o e-mail cadastrado no Web GAAT. A opção está disponível a partir do link *Forgot your password* (Figura 11). A tela relativa a recuperação de senha, *Password Reset,* pode ser observada na Figura 13.

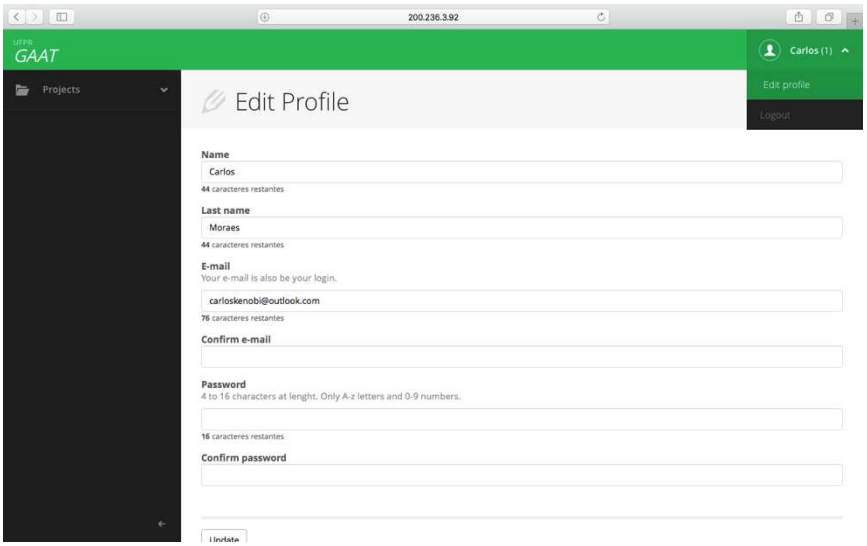
FIGURA 13 - TELA PARA RECUPERAÇÃO DE SENHA



FONTE: O autor (2016).

Após o cadastro o usuário pode acessar o Web GAAT utilizando e-mail e senha informados. Uma vez autenticado no sistema, o usuário pode editar seus dados clicando no menu presente no canto superior direito da tela, representado pela Figura 14. Neste caso as mesmas regras de validação para o registro de usuários são aplicadas.

FIGURA 14 - TELA PARA EDIÇÃO DOS DADOS CADASTRAIS



FONTE: O autor (2016).

Após o *logon*, os dados do usuário são armazenados na sessão, ficando disponíveis para serem utilizados por outros módulos.

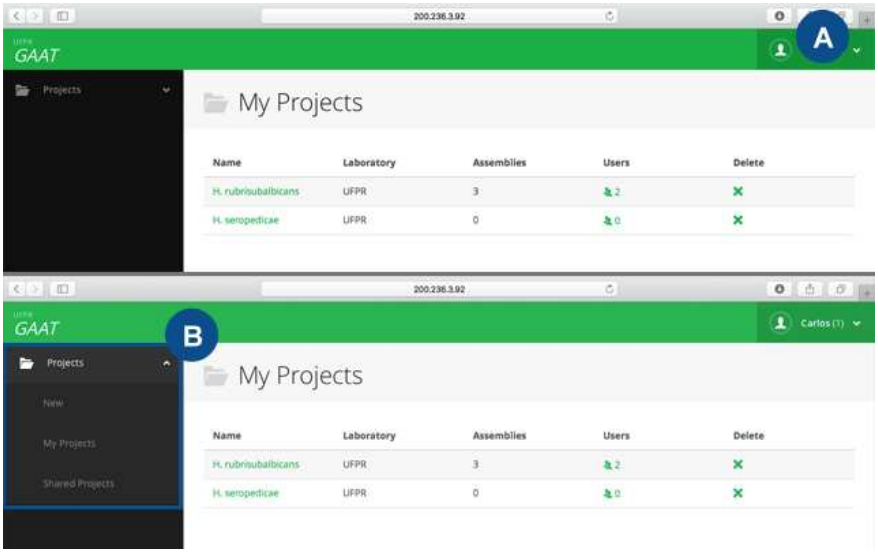
3.4.2 Project

O módulo Project foi criado para que os usuários possam gerenciar seus projetos de análise e anotação de dados sequenciados. Este módulo compreende as seguintes funcionalidades:

- Criação ou exclusão de projetos;
- Gerenciamento de usuários que podem acessar um projeto (compartilhamento);
- Gerenciamento do nível de acesso dos usuários a um determinado projeto.

O modulo é acessado imediatamente após o *logon* do usuário. Na Figura 15 é possível observar em A sua tela inicial e em B o menu expandido do módulo, após ser clicado. Por meio deste menu é possível criar um novo projeto ou acessar os projetos existentes criados pelo usuário.

FIGURA 15 - TELA INICIAL DO MÓDULO PROJECT



FONTE: O autor (2016).

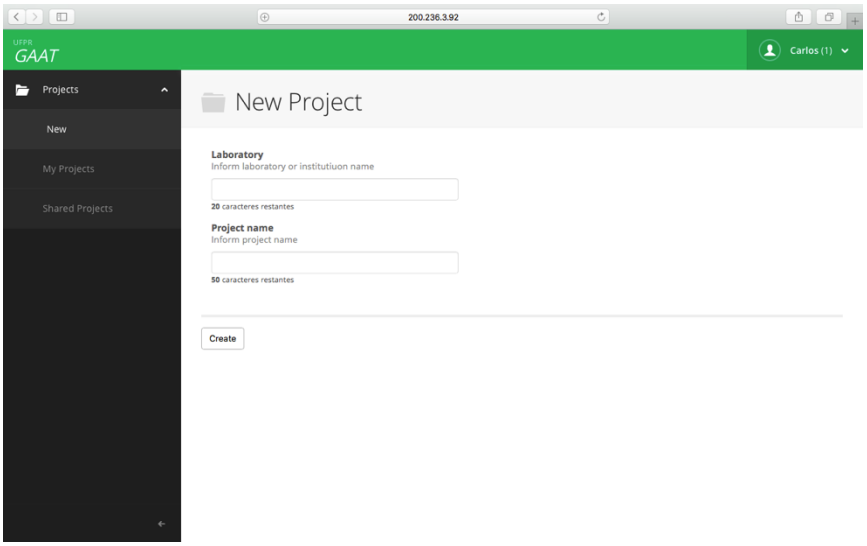
Para a criação de projetos, é necessário o preenchimento dos campos Laboratory e Name, de acordo com as seguintes especificações:

- Laboratory: Possui caráter informacional, para identificação de qual laboratório ou instituição determinado projeto pertence. É de preenchimento obrigatório e pode possuir no máximo 20 caracteres alfanuméricos, podendo conter espaços entre palavras.
- Name: Campo obrigatório destinado ao nome do projeto pode possuir no máximo 20 caracteres alfanuméricos, podendo conter espaços ou pontos. Deve ser único entre todos os projetos criados pelo usuário.

A tela para criação de projetos, visualizada na Figura 16, pode ser acessada a partir do item New do menu Projects.

Assim como no registro de usuários, as regras aplicadas para criação de projetos são validadas utilizando métodos fornecidos pelo framework CodeIgniter. Caso alguma regra não seja atendida, o software notificará o usuário sobre as correções que devem ser realizadas.

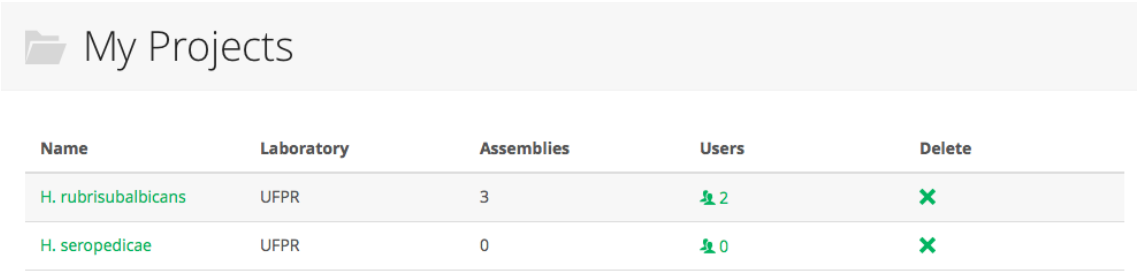
FIGURA 16 - TELA PARA CRIAÇÃO DE PROJETOS



FONTE: O autor (2016).

A vinculação de um projeto ao usuário é realizada com base nos dados do usuário presentes na sessão. Os projetos criados pelo usuário podem ser acessados pelo item My Projects do menu. Conforme tela visualizada na Figura 17, os projetos são listados alfabeticamente em uma tabela onde cada linha corresponde a um projeto. As colunas, por sua vez, contêm informações relativas ao projeto e links para que ações específicas sejam executadas pelo usuário.

FIGURA 17 – DETALHAMENTO DA LISTAGEM DE PROJETOS



FONTE: O autor (2016).

Conforme detalhado na Figura 17, as seguintes informações e ações estão presentes na listagem de projetos:

- Name: Contém o nome do projeto. Ao clicar no item o usuário é remetido para a tela contendo os dados sequenciados enviados para o projeto.
- Laboratory: Nome do laboratório;
- Assemblies: Quantidade de arquivos com dados sequenciados enviados para o projeto;
- Users: Link que direciona para a tela de gerenciamento de usuários com acesso ao projeto. Caso exista um compartilhamento, o item exibirá também a quantidade de usuários com acesso.
- Delete: Link para exclusão do projeto.

O compartilhamento de projetos compreende a solução proposta para que vários usuários possam trabalhar na anotação de dados sequenciados de um organismo, tornando a atividade colaborativa. Considerando que equipes podem ser formadas por colaboradores com diferentes atribuições, o compartilhamento prevê dois níveis de acesso:

- View: Destinado a usuários que podem acessar todos as informações de um projeto, mas que não podem anotar dados sobre eles;
- View and Edit: Destinado a usuários que irão atuar realizando anotação sobre os dados sequenciados.

A Tabela 1 apresenta as permissões e restrições relacionadas a cada papel disponibilizado. Na tabela é possível observar a também a existência do papel *Owner*,* sendo esta a definição atribuída ao usuário que criou o projeto.

TABELA 1 - PERMISSÃO E RESTRIÇÕES DOS PAPÉIS PARA USUÁRIO DISPONÍVEIS NO MÓDULO PROJECT

Papel	Gerenciar usuários	Enviar montagens	Excluir montagens	Visualizar montagens	Anotar dados	Excluir ORFs/Genes
Owner*	Sim	Sim	Sim	Sim	Sim	Sim
View/Edit	Não	Sim	Não	Sim	Sim	Sim
View	Não	Não	Não	Sim	Não	Não

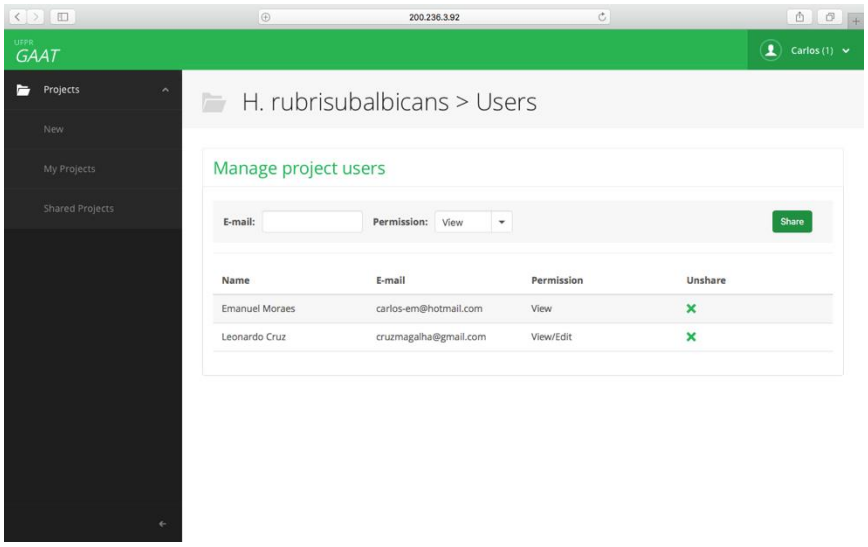
FONTE: O autor (2016).

Para que um projeto possa ser compartilhado, os seguintes passos são necessários:

1. A partir da listagem de projetos, Figura 17, clicar no ícone presente na coluna Users;
2. No campo E-mail informar o e-mail de um usuário cadastrado no Web GAAT;
3. Em Permission informar o nível de acesso desejado para o usuário.
4. Clicar em Share.

Caso o e-mail informado pertença a um usuário registrado no Web GAAT, o compartilhamento será realizado. Caso contrário o sistema informará sobre o erro. Os passos 2 a 3 podem ser observados na Figura 18, na qual é possível ainda visualizar todos os usuários com acesso ao projeto.

FIGURA 18 - TELA PARA GERENCIAMENTO DO COMPARTILHAMENTO DE UM PROJETO

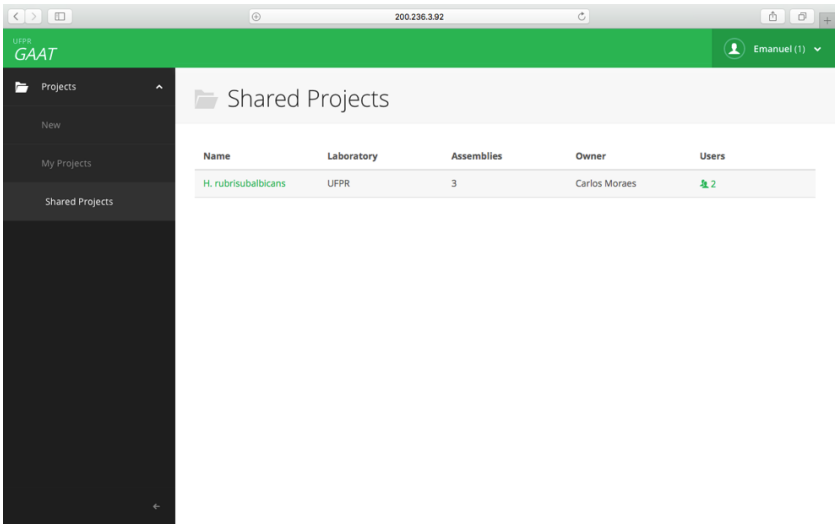


FONTE: O autor (2016).

Para remover um usuário do compartilhamento é preciso clicar no ícone presente na coluna Unshare, visualizado na Figura 18.

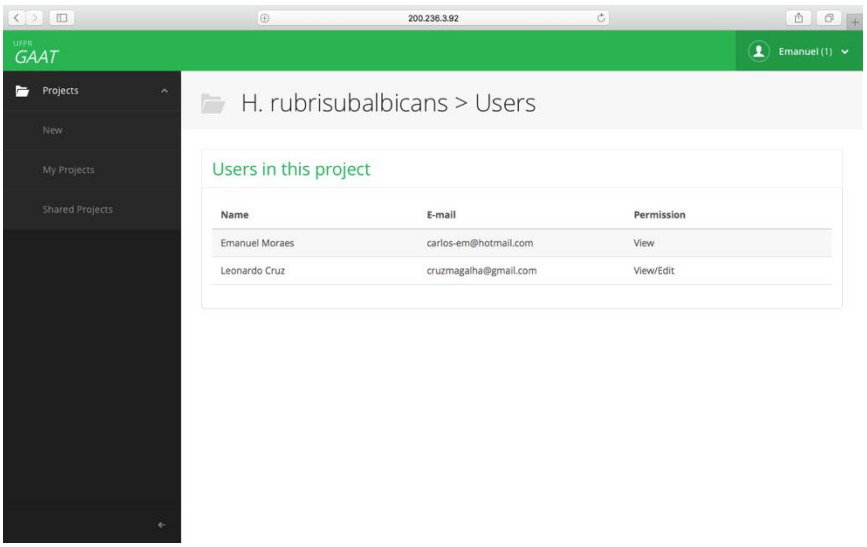
Por fim, o item Shared Projects, do menu, lista todos os projetos que foram compartilhados com o usuário logado, conforme Figura 19. Esta tela diferencia-se por não apresentar o ícone para exclusão de projetos, a opção é disponibilizada apenas para o dono do projeto, indicado na coluna *Owner*. Em projetos compartilhados, ao clicar no ícone relativo à coluna *Users* são listados todos os usuários presentes no compartilhamento, conforme Figura 20.

FIGURA 19 - TELA LISTANDO OS PROJETOS COMPARTILHADOS COM O USUÁRIO



FONTE: O autor (2016).

FIGURA 20 - TELA LISTANDO DEMAIS USUÁRIOS COM ACESSO AO PROJETO



FONTE: O autor (2016).

3.4.3 Assembly

Nomeado Assembly por ser destinado ao envio de montagem de genomas para anotação, o módulo é constituído por um conjunto de scripts e programas objetivados a fornecer os principais recursos utilizados no processo de análise e anotação de dados sequenciados sobre genomas. É dependente do módulo Project pois utiliza as funcionalidades deste para possibilitar o trabalho colaborativo e organização dos dados de um organismo em um mesmo projeto.

O módulo Assembly compreende as seguintes funcionalidades:

- Identificação automática de ORFs, tRNAs e predição de genes;
- Visualização de genomas de anotação sobre dados obtidos;
- Conversão automática de sequências de nucleotídeos em sequência de proteínas;
- Possibilidade de envio de sequencias para BLASTN e BLASTP;
- Armazenamento de histórico sobre dados anotados.

Neste módulo também foi aplicado o modelo de Visualização de Informação apresentado por Card et al. (1999). A Transformação de Dados ocorre durante a identificação automática de ORFs e predição de genes. Todos os dados encontrados são salvos no banco e organizados de maneira estruturada em tabelas. Seguindo o modelo apresentado pelo autor, cada linha da tabela representa um elemento, relacionado a uma montagem enviada por um usuário

As propriedades de cada elemento encontrado são distribuídas em colunas, de acordo com suas características. A abordagem permite que a visualização disponibilizada pelo módulo organize mais facilmente a informação a fim de representa-la graficamente e apresentar dados estatísticos sobre a montagem. O Mapeamento Visual ocorre posteriormente, quando o usuário seleciona dados para visualização e anotação, onde itens gravados no banco são distribuídos em uma estrutura visual para interação do usuário (Transformações Visuais).

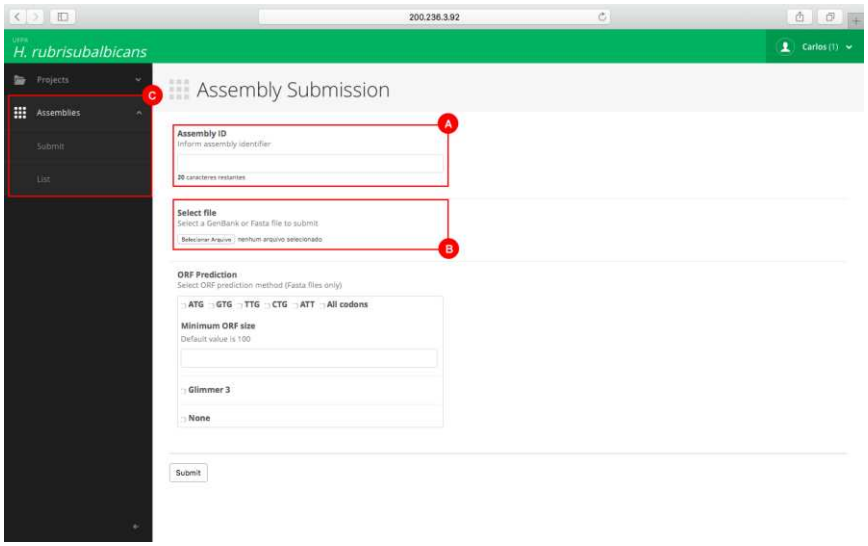
Para um melhor entendimento do fluxo de funcionamento do módulo, bem como dos scripts, programas envolvidos e técnicas de VI utilizadas no processo, as funcionalidades citadas anteriormente estão descritas nos tópicos a seguir.

3.4.3.1 Envio de montagens

Com o objetivo de desenvolver um software de fácil utilização, este processo foi desenhado para exigir o mínimo de configurações por parte do usuário. Possuindo apenas uma interface, para o envio do arquivo, o fluxo criado é automatizado e divido em subtarefas que proporcionam maior flexibilidade para manutenção e futuros desenvolvimentos.

A primeira parte do fluxo é referente ao envio do arquivo pelo usuário. A interface pode ser visualizada na Figura 21. Nesta tela o usuário deve informar um valor para o campo Assembly ID (A) e selecionar o arquivo a ser enviado (B):

FIGURA 21 - TELA REFERENTE AO ENVIO DE ARQUIVOS FASTA E GENBANK



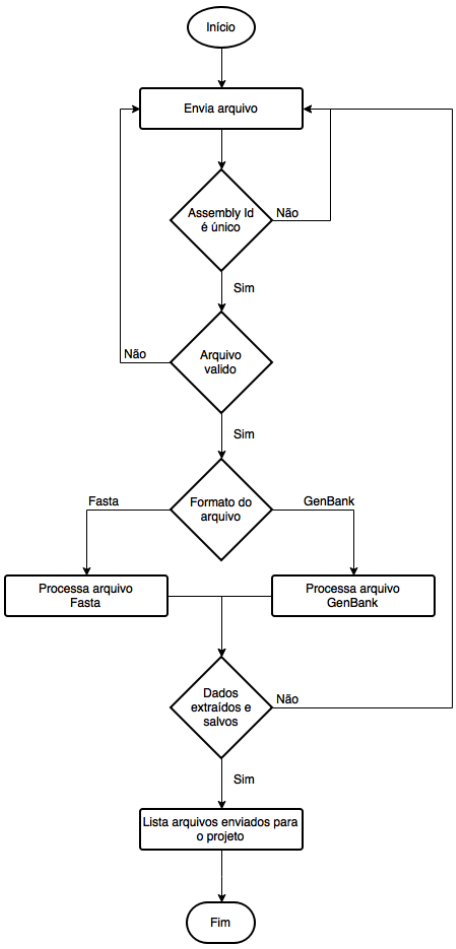
FONTE: O autor (2016).

Também é possível observar na Figura 21 o menu Assemblies (C) e suas opções: Submit para enviar arquivos e List para listar arquivos enviados. Como o módulo Assembly é dependente de Projects, o menu é disponibilizado para o usuário somente após a seleção de um projeto existente, conforme telas apresentadas nas figuras 17 e 19.

Após o envio do arquivo, o Web GAAT valida os dados informados e executa condicionalmente outras duas tarefas, uma para o caso de arquivos GenBank e outra para arquivos Fasta. Essa divisão proporciona não somente tratamento separado para cada formato, mas atende também aos requisitos de fácil atualização e manutenção, pois alterações no processo relacionado a um tipo de arquivo não afetam o outro. A abordagem permite ainda que outros formatos de arquivo sejam aceitos no futuro sem que interfiram no processo atual.

Uma visão geral do fluxo, com a divisão das tarefas, pode ser observada na Figura 22. Note que tanto o Assembly ID informado, quanto o tipo de arquivo são validados e, caso não passem pela validação, o sistema retorna à tela de envio de arquivo notificando o usuário sobre o erro. Também é possível visualizar os processos condicionais relativos ao tipo de arquivo enviado.

FIGURA 22 - DIAGRAMA DE FLUXO DO PROCESSO DE ENVIO E EXTRAÇÃO DE ARQUIVOS

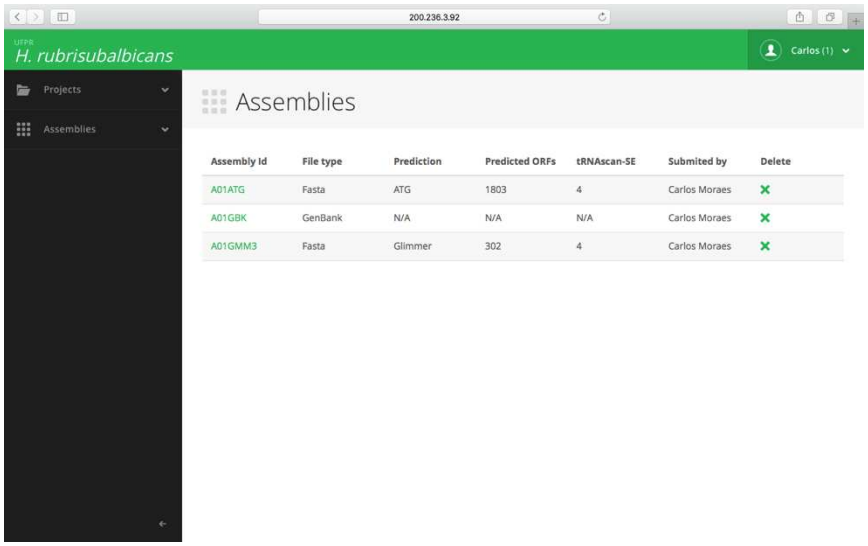


FONTE: O autor (2016).

Para o tratamento de arquivos GenBank foi criado um script em Python que, utilizando *parsers* da biblioteca BioPython, percorre todo o arquivo extraindo suas *features* e gravando no banco informações referentes ao tipo da *feature*, posição na sequência e seus *qualifiers*. Para arquivos Fasta um script em Perl analisa as sequências e cria, para cada *contig* presente, um arquivo Fasta separado, fluxo este similar ao tratamento dado pelo GAAT aos arquivos deste tipo. O processo é realizado desta forma para que os outros scripts do programa, relacionados à predição de genes e busca por ORFs e tRNAs sejam propriamente executados. Esta abordagem é melhor explicada no tópico 3.4.3.2, referente à busca automática de ORFs, tRNAs e predição de genes.

Após o envio, os arquivos são listados na tela apresentada na Figura 23, sob o nome de Assemblies.

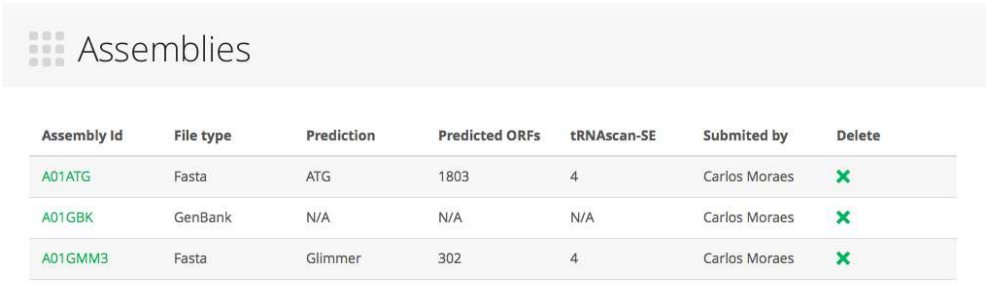
FIGURA 23 - LISTAGEM DE ARQUIVOS ENVIADOS (ASSEMBLIES)



FONTE: O autor (2016).

Na listagem da Figura 23 são exibidas informações referentes ao processo de envio, busca por ORFs, tRNAs e predição de genes, conforme detalhado na Figura 24.

FIGURA 24 - DETALHAMENTO DA LISTAGEM DE ASSEMBLIES



FONTE: O autor (2016).

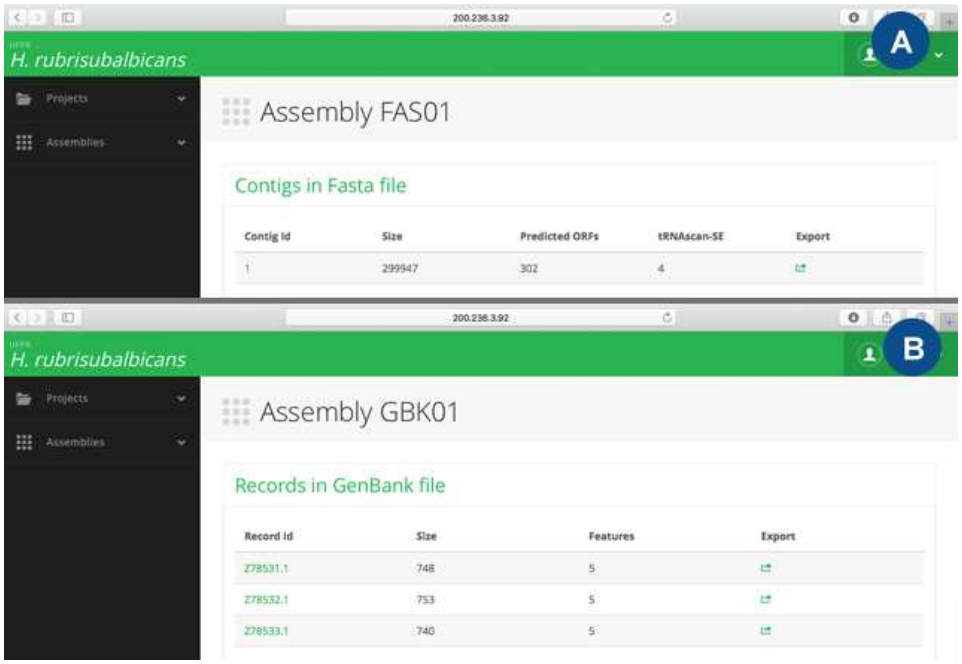
Conforme observado na Figura 24, são exibidas na tela de listagem as seguintes informações:

- Assembly ID: ID definido pelo usuário no momento do envio;
- File type: Tipo do arquivo enviado;
- Prediction: Método escolhido para busca automática por ORFs;
- Predicted ORFs: Número de ORFs encontradas;
- tRNAscan-SE: Número de tRNAs encontrados pelo tRNAscan-SE;
- Submited by: Usuário que enviou a sequência, importante para identificar quem realizou o envio em caso de projeto compartilhado.

- Delete: Opção para excluir os arquivos enviados, disponível apenas para o dono do projeto.

Ao clicar em um item presente na coluna Assembly ID, conforme arquivo enviado para o programa, o usuário é remetido à tela contendo a listagem de contigs (FASTA, Figura 25-A) ou de sequence records (GenBank, Figura 25-B).

FIGURA 25 - LISTAGEM DE CONTIGS OU SEQUENCE RECORDS EXISTENTES NOS ARQUIVOS ENVIADOS



FONTE: O autor (2016).

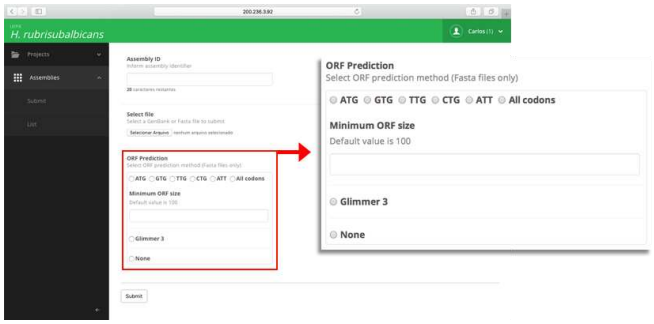
Conforme a Figura 25, na listagem de *contigs* é possível verificar o tamanho da sequência (Size), quantidade de ORFs (Predicted ORFS) e tRNAs (tRNAscan-SE) encontrados. Para *sequence records*, além do tamanho da sequência, é informada a quantidade de *features* encontradas no arquivo. Também está presente nesta listagem a opção Export, destinada a exportar todo os dados anotados em um arquivo GenBank.

3.4.3.2 Busca automática por ORFs, tRNAs e predição de genes

Identificar ORFs ao longo de sequências contendo milhares de pares de base pode ser considerada uma tarefa praticamente impossível ao olho humano. Seria necessário buscar ao longo dos arquivos de sequência por regiões com um códon de início e imediatamente procurar pelo códon de término, repetindo

a tarefa para cada fase de leitura. Integrado ao Web GAAT existem duas opções para a busca por ORFs, tornando o processo automatizado: busca a partir de um códon de início e busca utilizando o software GLIMMER. A escolha por um dos métodos é disponibilizada para o usuário no momento do envio do arquivo, conforme Figura 26.

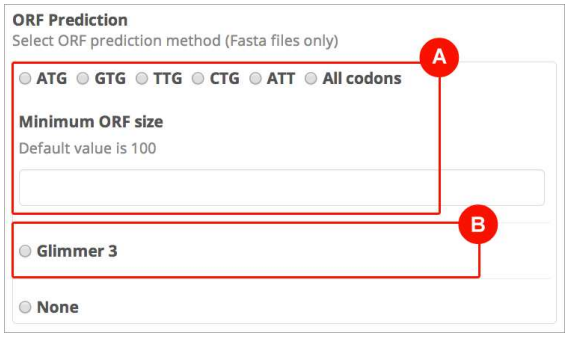
FIGURA 26 - OPÇÕES PARA BUSCA AUTOMÁTICA POR ORFS



FONTE: O autor (2016).

Os métodos de busca estão detalhados na Figura 27, onde em A é possível visualizar as opções referentes a busca por códons e em B a opção de utilizar o software GLIMMER.

FIGURA 27 - MÉTODOS DISPONÍVEIS PARA BUSCA POR ORFS.

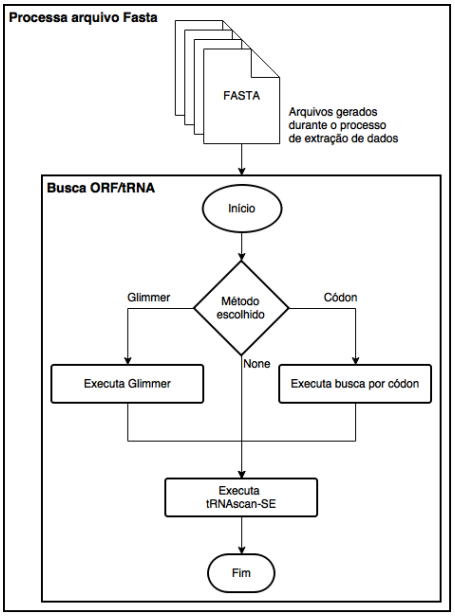


FONTE: O autor (2016).

Assim como o restante da aplicação, a busca por ORFs também foi desenhada para ser flexível. Na Figura 28 é possível visualizar os processos referentes a cada método ocorrendo de forma condicional. No Web GAAT, scripts separados são responsáveis por cada opção, possibilitando modificações pontuais. O mesmo ocorre para a busca por tRNAs, que é realizada pelo

software tRNAscan-SE; um script separado é responsável pela execução do programa e tratamento de seus resultados.

FIGURA 28 - FLUXO DO PROCESSO DE BUSCA POR ORFS E TRNAS



FONTE: O autor (2016).

É possível observar ainda na Figura 28 que a busca por ORFs e tRNAs ocorre durante o envio de arquivos FASTA, pois neste momento os *contigs* são separados e gravados em arquivos distintos. A abordagem é necessária uma vez que os softwares GLIMMER e tRNAscan-SE (utilizado para buscar tRNAs) só aceitam arquivos como formato de entrada de dados.

Para o processo referente à busca por códons, um script em Perl percorre as sequências de nucleotídeos presentes nos arquivos procurando pelo códon de início escolhido e, ao encontra-lo, procura imediatamente por um códon de término. Caso a ORF encontrada tenha um tamanho igual ou maior ao valor informado no campo *Minimum ORF Size* (Figura 27-A), os dados são gravados no banco para que possam ser visualizados e anotados. O processo é repetido para cada frame de leitura do DNA.

Caso o usuário opte por utilizar o software GLIMMER, os dados gerados pelo programa são interpretados por outro em Perl que extrai as informações obtidas e grava no banco os resultados.

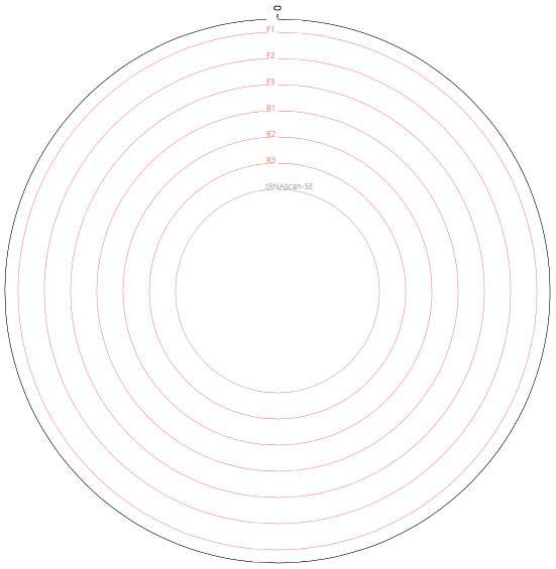
Assim como no GAAT, independentemente do método escolhido, o Web GAAT cria, para cada ORF encontrada, um arquivo Fasta contendo a sequência de nucleotídeos e outro contendo a sequência de proteínas. Estes dados podem ser utilizados em programas externos, como BLASTN e BLASTP, por exemplo.

Com relação à busca por tRNAs, esta é realizada automaticamente pelo Web GAAT. Um script em Perl é responsável por executar o programa tRNAscan-SE e analisar seus resultados. Após a análise os dados referentes à localização do tRNA, tipo, seu anticódon e score são gravados no banco.

3.4.3.3 Visualização de dados sequenciados

Buscando oferecer ao usuário a possibilidade de analisar uma região completa de um genoma e seguindo o modelo apresentado por (CARD et al., 1999) e definições de (LIMA, 2011), foi construído um eixo nominal circular, subdivido em sete regiões, conforme exibido na Figura 29.

FIGURA 29 - EIXO NOMINAL PARA VISUALIZAÇÃO DE GENOMAS

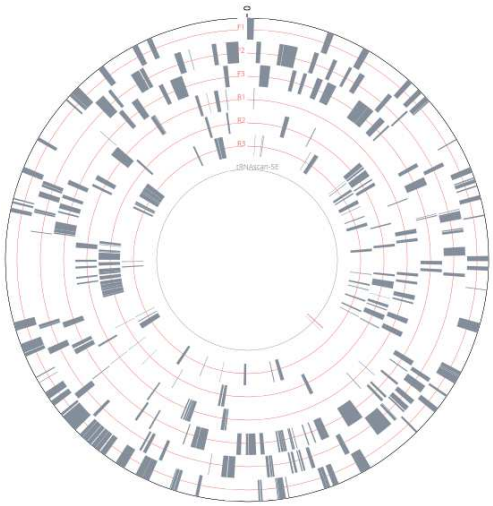


FONTE: O autor (2016).

Na visualização proposta (Figura 29), cada região do eixo é definida por círculos, que representam cada um dos seis frames de leitura (F1, F2 e F3, R1, R2 e R3). Um sétimo círculo, ao centro foi criado para apresentar os dados referentes aos resultados do programa tRNAScan-SE. A proposta tem como objetivo oferecer o panorama geral de toda uma região do genoma. No modelo,

a distribuição de ORFs, genes preditos e tRNAs ocorre sequencialmente de acordo com sua posição no *frame*, como se estes fossem eixos ordenados. Cada elemento no eixo, possui um ponto de início e fim, possibilitando a relação de tamanho entre os itens e de cada item com o tamanho da região. A abordagem também possibilita a fácil identificação de regiões com maior número de *features.* Um exemplo pode ser observado na Figura 30, onde elementos em cinza representam ORFs ou genes preditos e elementos em vermelho tRNAs encontrados.

FIGURA 30 - DISTRIBUIÇÃO DE ORFS E TRNAS AO LONGO DO PADRÃO DE VISUALIZAÇÃO

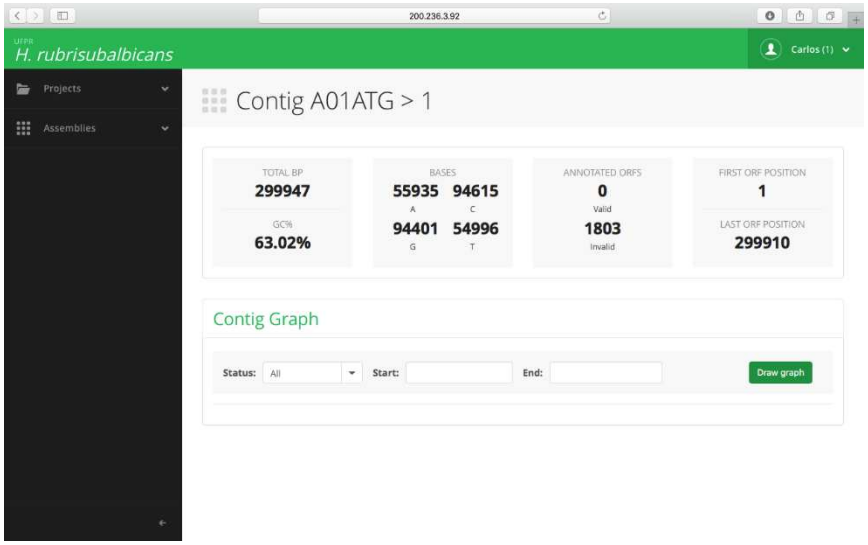


FONTE: O autor (2016).

O gráfico é gerado a partir da seleção de um *contig* ou de um *sequence record*, conforme telas da Figura 25. Para o caso de *contigs*, após a seleção, o usuário é remetido para a tela apresentada na Figura 31, onde é possível verificar algumas estatísticas relacionadas à sequência:

- Total BP: Total de pares e base;
- GC%: percentual de GC;
- Bases: número de bases A, C, T e G;
- Annotated ORFs: número de ORF anotadas, divididas em válidas e inválidas;
- First ORF Position: posição inicial da primeira ORF encontrada;
- Last ORF Position: posição final da última ORF encontrada.

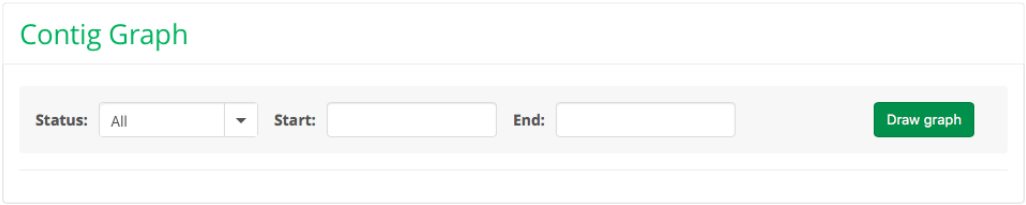
FIGURA 31 - VISUALIZAÇÃO DE UM CONTIG



FONTE: O autor (2016).

Na mesma tela da Figura 31, logo abaixo dos dados estatísticos, é possível observar um quadro com o título Contig Graph. Neste quadro, detalhado na Figura 32, estão os filtros que devem ser utilizados para desenhar o gráfico contendo as ORFs e tRNAs encontrados: Status, Start e End.

FIGURA 32 - FILTROS PARA DESENHO DO GRÁFICO CONTENDO ORFs e tRNAs

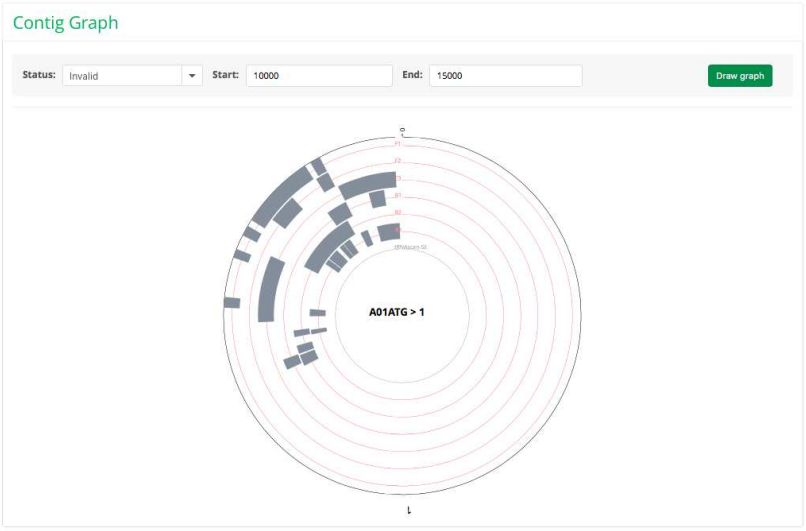


FONTE: O autor (2016).

O campo status é utilizado para filtrar quais ORFs devem ser exibidas: todas, apenas válidas ou inválidas. Os campos Start e End servem para que o usuário limite a porção do *contig* a ser exibida. Os valores devem estar entre os números apresentados nos dados estatísticos de First ORF Position e Last ORF Position. Um exemplo de aplicação dos filtros pode ser observado na Figura 33, na qual é possível visualizar todas as ORFs inválidas presentes entre as posições 10000 e 15000 pares de base do *contig*. Ao posicionar o mouse sobre

um dos elementos é possível visualizar mais informações sobre ele, conforme Figura 34.

FIGURA 33 - GRÁFICO DESENHADO APÓS APLICAÇÃO DOS FILTROS

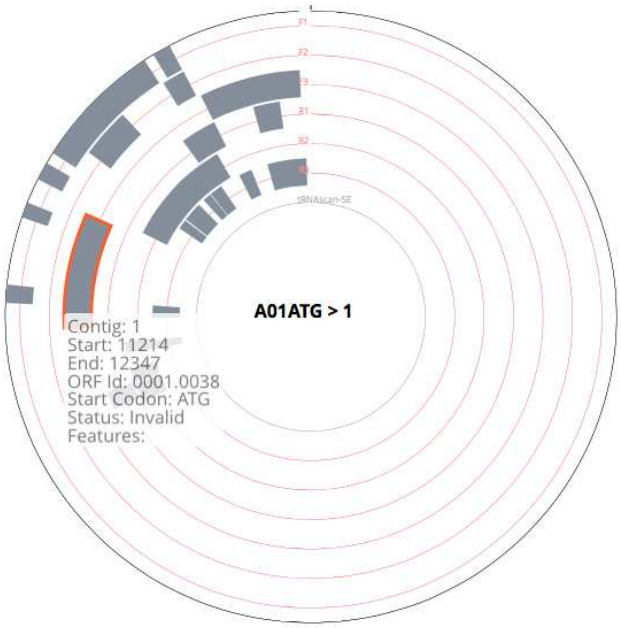


FONTE: O autor (2016).

É possível visualizar também na Figura 34 uma das ORFs marcadas em laranja, indicando que o ponteiro do mouse foi posicionado sobre o elemento. Com esta ação são exibidas as seguintes informações sobre a ORF:

- Contig: Número de identificação do contig referente à ORF;
- Start: Posição inicial da ORF;
- End: Posição final da ORF;
- ORF ID: Número de identificação da ORF;
- Start Codon: Códon de início da ORF;
- Status: Status da ORF, válida ou inválida;
- Features: Propriedades da ORF com seus respectivos *qualifiers*.

FIGURA 34 - INFORMAÇÕES EXIBIDAS AO PASSAR O MOUSE POR CIMA DE UMA ORF

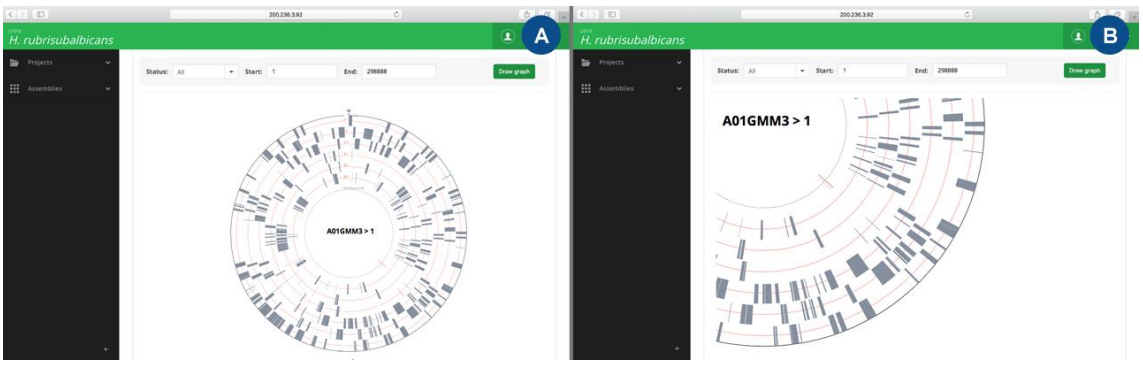


FONTE: O autor (2016).

Com exceção dos campos Contig, Start, End, ORF ID e Start Codon, todos os outros campos são relacionados à etapa de anotação, que será apresentada na sessão 3.4.3.4 deste trabalho.

Para melhor visualizar regiões específicas do gráfico desenhado, o Web GAAT permite a utilização de *zoom*. A aplicação do recurso pode ser observada na Figura 35. Note na figura em (A) a região de um genoma com seus respectivos elementos e em (B) a aplicação de zoom em uma porção específica da sequência. O zoom é aplicado utilizando o scroll do mouse.

FIGURA 35 - APLICAÇÃO DE ZOOM PARA VISUALIZAÇÃO DE UMA REGIÃO ESPECÍFICA

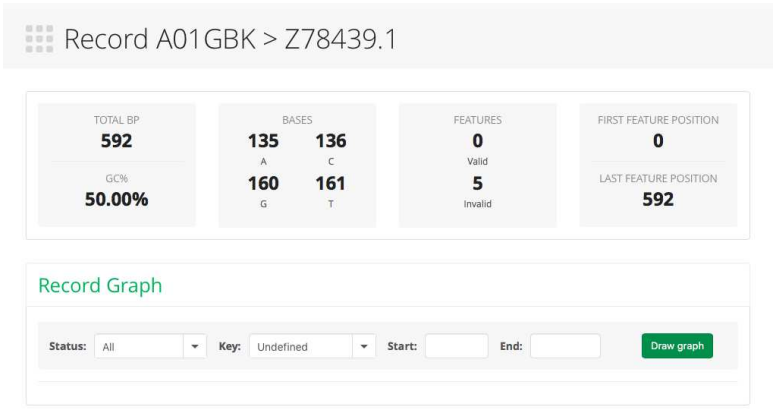


FONTE: O autor (2016).

A metodologia aplicada segue a definição de (LIMA, 2011), na qual inicialmente é apresentado um panorama geral com as propriedades básicas de cada elemento que, no presente contexto, são posição e tamanho de cada elemento em seu respectivo eixo ordenado. A partir do interesse do usuário é possível visualizar mais informações sobre um determinado item ou visualizar com maior proximidade uma região de interesse.

Para o caso de *sequence records*, o filtro para desenho do gráfico é ligeiramente diferente, conforme Figura 36, onde é observado a existência de um filtro a mais, Key. O campo foi adicionado a fim de possibilitar a exibição de features específicas existentes em arquivos GenBank.

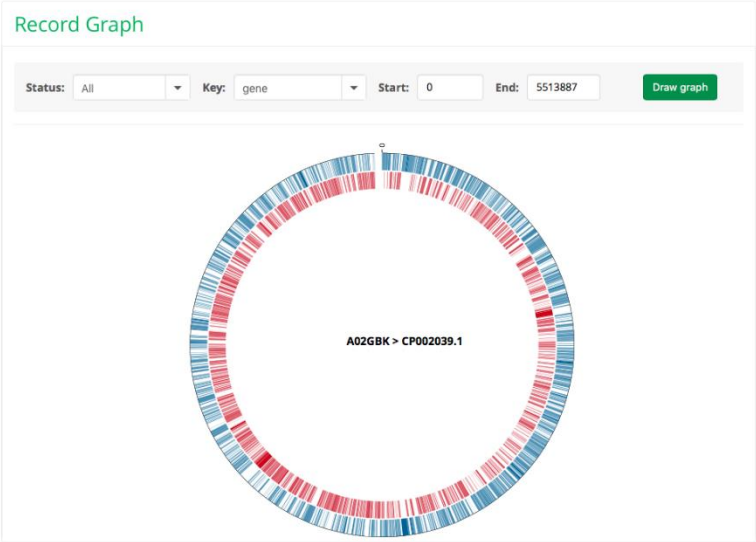
FIGURA 36 - FILTROS EXISTENTES PARA SEQUENCE RECORDS



FONTE: O autor (2016).

A visualização de *sequence records* também possui diferenças em relação ao modelo aplicado para os *contigs*. Ao invés dos seis círculos, apenas dois são utilizados, para representar as posições relativas a *upstream* e *downstream*, conforme Figura 37.

FIGURA 37 - VISUALIZAÇÃO DE GENES EM UM SEQUENCE RECORD

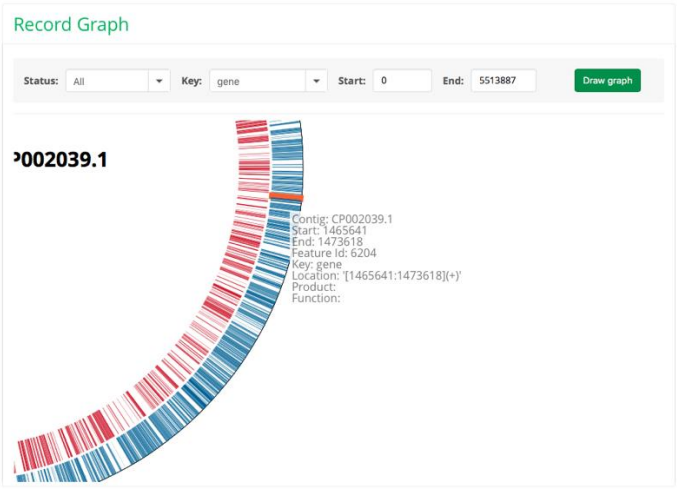


FONTE: O autor (2016).

É possível observar na Figura 37, após utilização do filtro, todos os genes presentes entre 0 e 5513887, tamanho total da sequência. Os elementos referentes à posição *upstream* são desenhados no círculo externo (azul). Em vermelho estão desenhados os genes encontrados na posição *downstream*. Assim como as ORFs e tRNAs, cada elemento foi distribuído considerando sua localização no genoma.

Na Figura 38 é possível observar a utilização de zoom também para *sequence records*.

FIGURA 38 - APLICAÇÃO DE ZOOM EM UM SEQUENCE RECORD



FONTE: O autor (2016).

Nota-se na Figura 38 que a aplicação de zoom é útil para analisar regiões do genoma com maior concentração de elementos.

Todas as funcionalidades relacionadas às interações do usuário com os gráficos foram desenvolvidas utilizando a biblioteca BioCircos. A utilização se dá pela padronização do código usado para desenho e tratamento das interações.

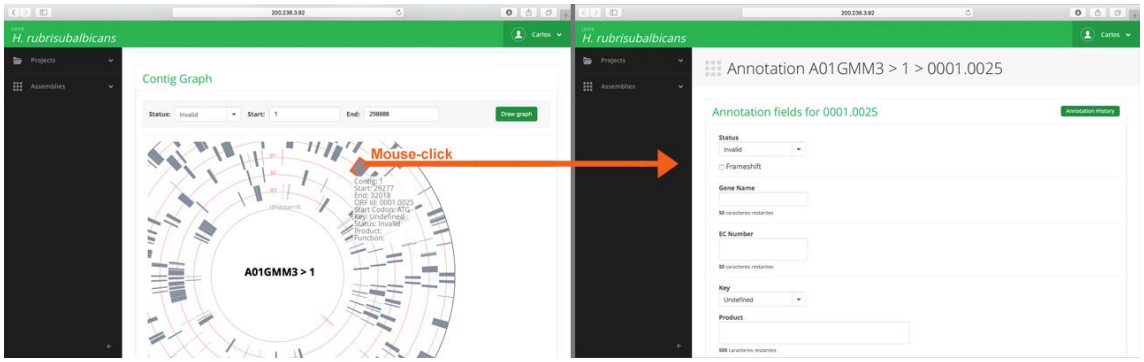
3.4.3.4 Anotação de dados sequenciados

A anotação de dados é disponibilizada pelo Web GAAT a partir da seleção de ORFs ou features desenhadas no gráfico de visualização. Após clicar em um dos elementos presentes no gráfico, o usuário é direcionado para uma tela onde um formulário permite que as seguintes informações sejam anotadas:

1. **Status**: Campo para que o usuário selecione se o elemento é válido ou inválido.
2. **Frameshift:** Campo para que o usuário selecione se o elemento possui ou não frameshift;
3. **Gene Name:** Campo para que o usuário informe o nome do gene;
4. **EC Number:** Campo para que o usuário informe o EC Number do elemento;
5. **Feature Key:** Campo para que o usuário selecione o tipo do elemento. Por padrão o valor é Undefined e estão disponíveis as opções Gene, CDS, misc_feature, source, tRNA e rRNA, sendo estas as mais comuns.
6. **Qaulifier Key:** Campo para que seja informado o tipo de propriedade (*qualifier*) relativo ao elemento;
7. **Qualifier Value:** Campo para que seja informado o valor referente a propriedade do elemento.
8. **Comments:** Campo para informações gerais ou comentários sobre o elemento.

O fluxo de navegação para acesso à tela de anotação pode ser observado na Figura 39.

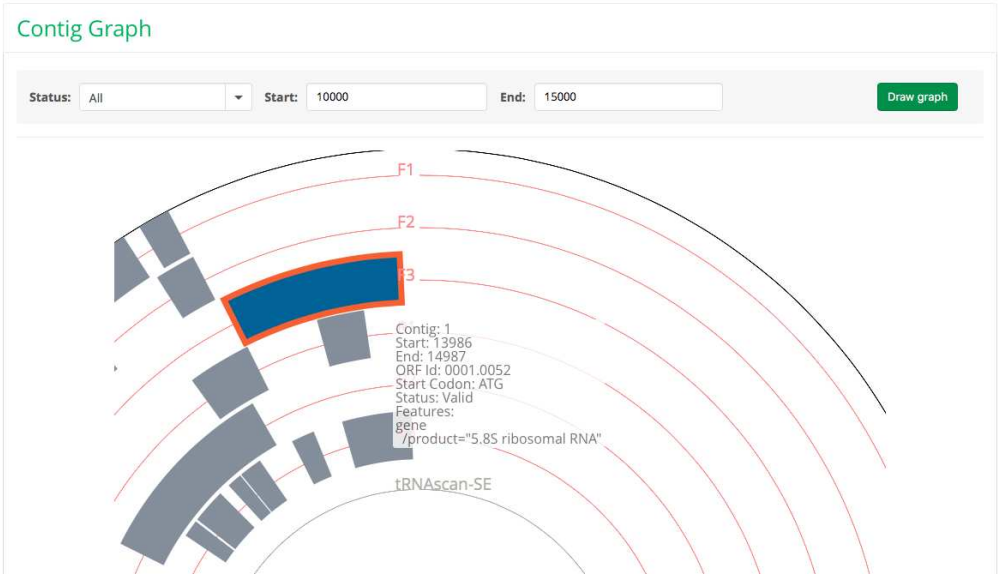
FIGURA 39 - NAVEGAÇÃO ENTRE TELAS DE VISUALIZAÇÃO E ANOTAÇÃO



FONTE: O autor (2016).

Após preencher os campos, os dados anotados podem ser visualizados no gráfico, conforme demonstrado na Figura 40.

FIGURA 40 - ALTERAÇÕES EM UMA ORF APÓS ANOTAÇÃO



FONTE: O autor (2016).

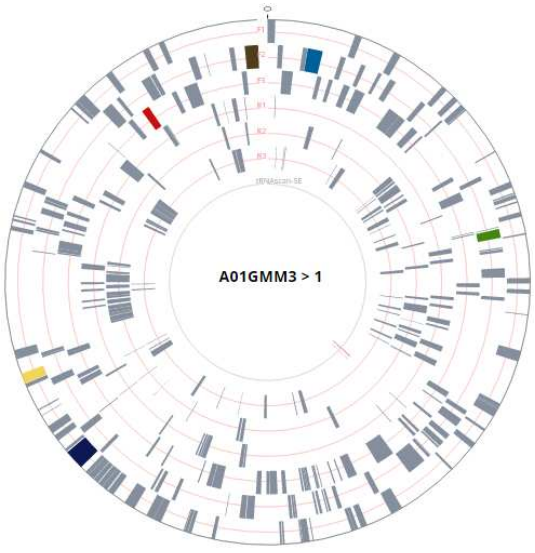
É possível visualizar na Figura 40 que, dentre os campos disponíveis para anotação, são exibidos no gráfico, ao posicionar o ponteiro do mouse sobre o elemento, as *features*. Conforme relatado no tópico 3.4.3.3, estas informações dependem da etapa de anotação para serem exibidas. Nota-se também que o elemento desenhado teve sua cor alterada de cinza para azul.

A utilização de cores é outro recurso implementado para a visualização de ORFs. Esta abordagem permite que o usuário identifique rapidamente quais elementos já foram anotados e de que tipo são; para cada valor presente no campo Key uma cor específica é atribuída ao respectivo elemento no gráfico, conforme lista a seguir:

- Azul: Gene;
- Azul Marinho: CDS;
- Verde: Source;
- Amarelo: misc_feature;
- Vermelho: tRNA;
- Marrom: rRNA;
- Azul claro: Quando há mais de uma *feature* na mesma posição.

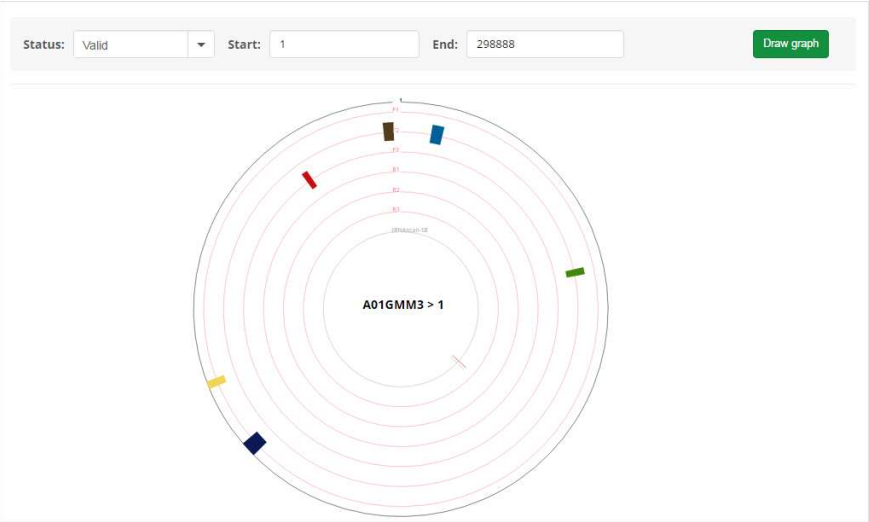
O método segue o modelo proposto por (CARD et al., 1999), onde a utilização de cor é uma propriedade gráfica que pode ser utilizada para caracterizar marcas visuais em determinado contexto. Um exemplo mais completo sobre a utilização de cor pode ser visualizado na Figura 41. Note na figura que a caracterização possibilita a distinção dos elementos com base em suas características. Utilizando o filtro para visualizar apenas ORFs válidas, o gráfico é gerado conforme Figura 42.

FIGURA 41 - UTILIZAÇÃO DE COR PARA DIFERENCIAÇÃO DE ELEMENTOS



FONTE: O autor (2016).

FIGURA 42 - APLICAÇÃO DE FILTRO PARA VISUALIZAÇÃO DE ELEMENTOS VÁLIDOS

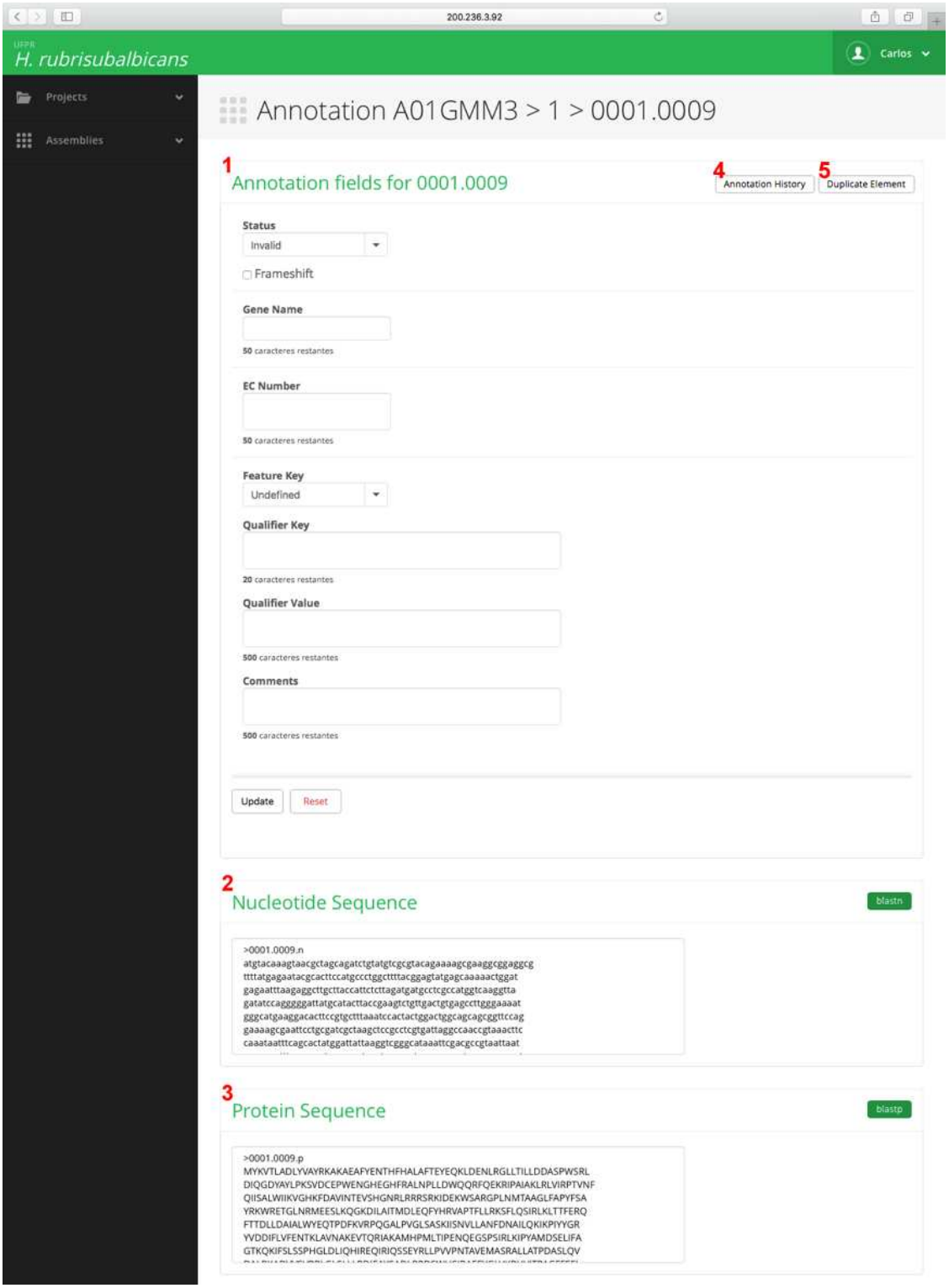


FONTE: O autor (2016).

Um panorama geral da tela de anotação pode ser observado na Figura 43. A tela é composta pelos campos para anotação (1) e, logo abaixo, as sequencias de nucleotídeos e aminoácidos são disponibilizadas para BLAST (2 e 3). No canto superior direito da tela está a opção para visualizar o histórico de anotação (4). Tanto BLAST como o histórico de anotação serão tratados nos tópicos a seguir.

Na Figura 43-5 é possível observar ainda a existência do item *Duplicate Element*. A partir desta opção é possível duplicar o elemento e atribuir outras *features*, caso necessário.

FIGURA 43 - PANORAMA GERAL DA TELA DE ANOTAÇÃO



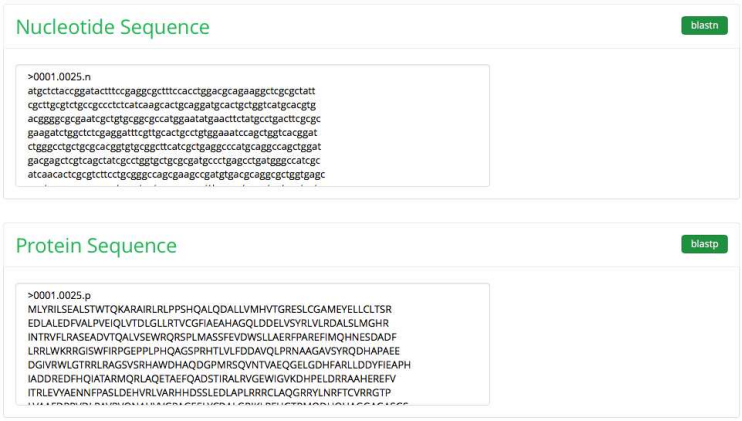
FONTE: O autor (2016).

3.4.3.5 BLASTN e BLASTP

Amplamente utilizado em pesquisas relacionadas à anotação de genomas, o BLAST tem como objetivo encontrar regiões de similaridade entre sequências, sendo fundamental para inferir relações funcionais e evolutivas das mesmas, bem como ajudar a identificar os membros das famílias de genes.

No Web GAAT as versões BLASTN e BLAST, utilizados para comparar respectivamente sequencias de nucleotídeos e aminoácidos, estão disponíveis para uso na tela de anotação de genomas (Figura 43-2 e 43-3), conforme detalhado na Figura 44.

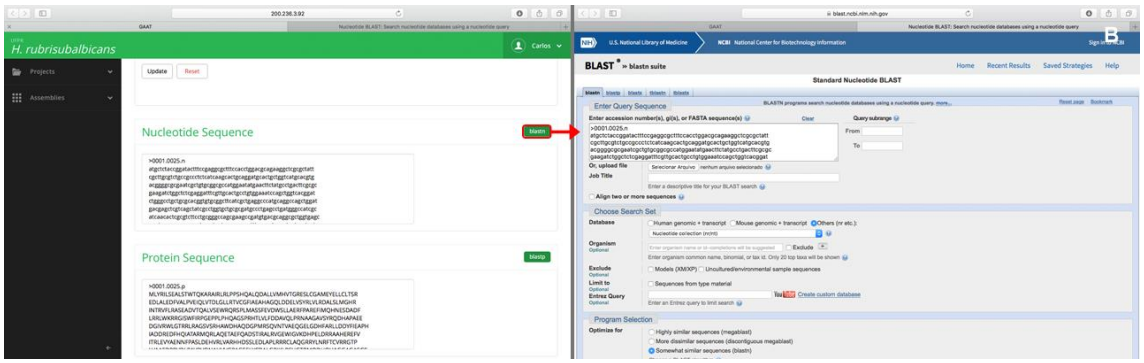
FIGURA 44 - OPÇÕES PARA BLASTN E BLASTP



FONTE: O autor (2016).

Visualizado na Figura 44, estão as opções disponibilizadas pelo Web GAAT para BLASTN e BLASTP. Ao selecionar uma das opções, a sequência é enviada para o site do NCBI, conforme exemplificado na Figura 45.

FIGURA 45 - EXEMPLO DE ENVIO PARA BLASTN



FONTE: O autor (2016).

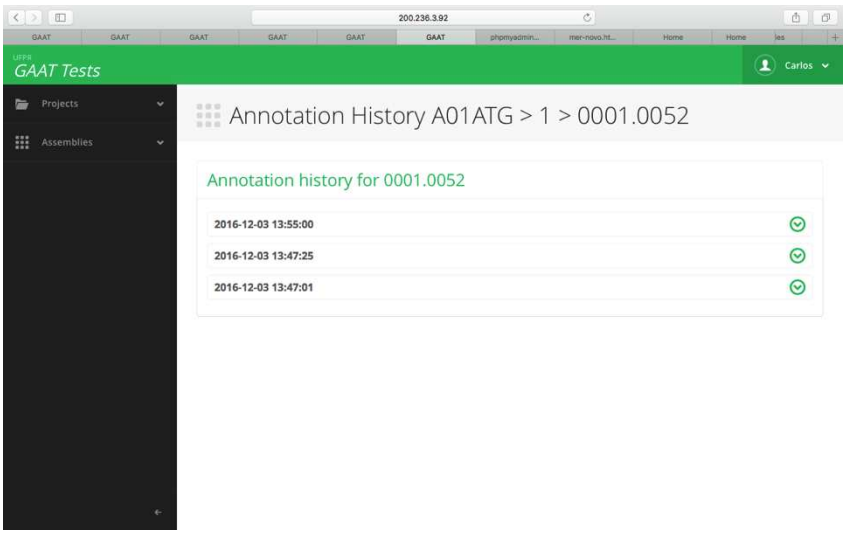
Conforme exemplo da Figura 45, ao clicar em uma das opções para BLAST, a respectiva sequência, referente à ORF que está sendo anotada, é enviada para o site do NCBI, onde o usuário pode executar o programa de acordo com as opções disponíveis.

3.4.3.6 Histórico de atualizações

O Web GAAT armazena em seu banco todas as alterações realizadas em ORFS e *features* na etapa de anotação. Este recurso é importante para manter um histórico daquilo que foi alterado em um elemento em determinado momento, principalmente em projetos compartilhados onde diferentes pesquisadores estão envolvidos tendo acesso aos mesmos dados. No histórico são exibidas todas as informações de uma ORF que tenha sido modificada, além do dia e hora de cada alteração e também quem realizou determinada alteração.

Para acessar o histórico de alterações é necessário clicar no botão "Annotation History", mostrado da Figura 43-4. Na Figura 46 é possível observar com detalhes as informações da tela de histórico.

FIGURA 46 - TELA DE HISTÓRICO DE ANOTAÇÕES

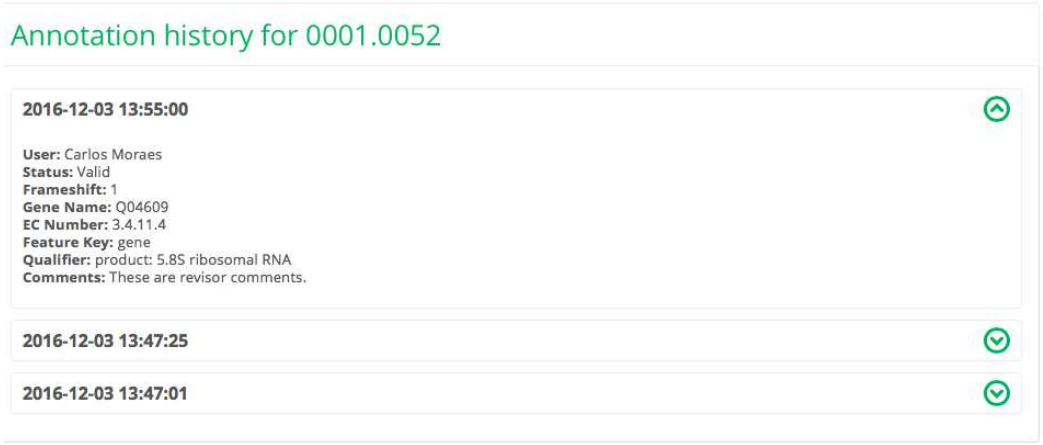


FONTE: O autor (2016).

Conforme observado na Figura 46, na tela são listadas por ordem de data e hora todas as anotações realizadas em um elemento. Ao clicar no ícone à direita de cada item da lista, é possível visualizar as anotações realizadas no dia

e hora indicados. Na Figura 47 é possível visualizar as informações disponíveis para cada item.

FIGURA 47 - INFORMAÇÕES DISPONÍVEIS NO HISTÓRICO DE UM ITEM ANOTADO



FONTE: O autor (2016).

Na Figura 47 é possível observar os itens anotados na data e hora especificados. Também é exibido o nome do usuário que realizou a anotação, informação bastante útil em projetos compartilhados, onde mais de um pesquisador pode anotar informações sobre um elemento.

3.5 DOCUMENTAÇÃO

Este tópico destina-se a apresentar os documentos de software relacionados ao Web GAAT. Inicialmente apresenta-se o MER, Modelo Entidade-Relacionamento. Por fim é apresentado o fluxograma do sistema, onde são especificadas as funcionalidades e entradas de dados.

3.5.1 Disponibilização e Instalação

Para facilitar a utilização do Web GAAT, uma máquina virtual (VM) contendo a ferramenta instalada e pronta para uso está disponível para download no endereço http://gaatweb.tecnologia.ws/webgaat-vm.zip.

Recomenda-se a utilização do programa VM VirtualBox (VIRTUALBOX, 2017) para execução da VM. Um atalho para execução do Web GAAT está presente na área de trabalho do sistema operacional.

Alternativamente é possível baixar o código fonte do Web GAAT para instalação em um servidor de preferência. Para isso um arquivo compactado (ZIP) foi disponibilizado no endereço http://gaatweb.tecnologia.ws/webgaat.zip. Após o download e descompactação é necessário seguir as instruções de instalação presentes no arquivo WebGAAT-Installation.txt, localizado no diretório raiz.

3.5.2 MER

O Modelo de Entidade-Relacionamento representa os objetos envolvidos no armazenamento de dados do Web GAAT e como estes se relacionam. De acordo com a divisão modular da plataforma, cada conjunto de tabelas é referente a um dos módulos: Login, Project ou Assembly.

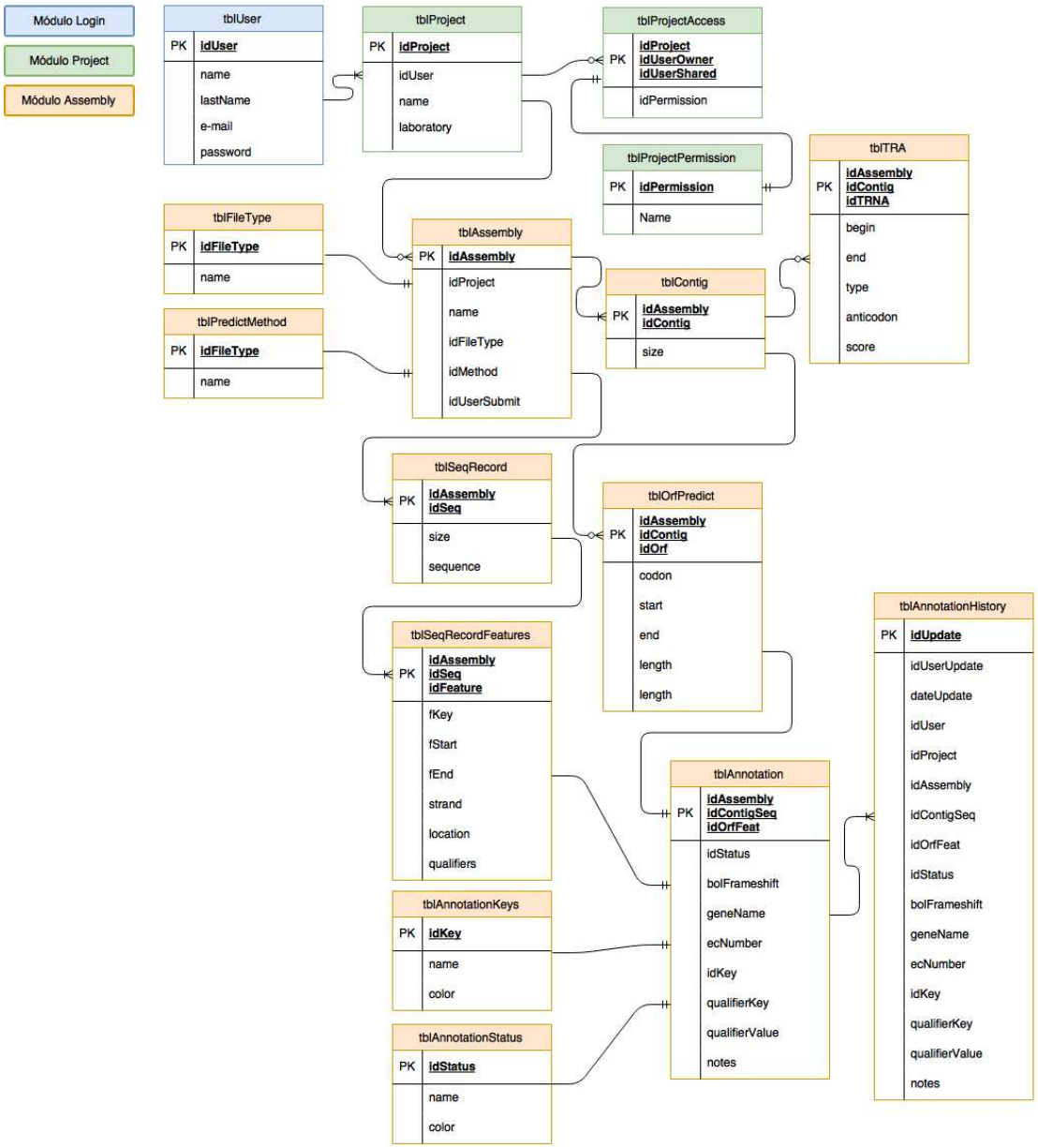
Para o módulo Login, apenas uma tabela foi criada, tblUser, nela são gravados todos os dados do usuário: nome, sobrenome, e-mail e senha. Na tabela também é gerado um identificador único para cada usuário, o campo idUser, o qual é utilizado para que projetos sejam associados a um usuário e também para realizar o compartilhamento de um projeto, indicando aqueles que podem acessa-lo.

Para o módulo Project, os objetos criados objetivam-se ao armazenamento de informações sobre um projeto e acesso ao mesmo, conforme tabelas tblProject e tblProjectAccess respectivamente. O campo idUser, existente em tblProject, é utilizado para associar o projeto ao usuário. Na tabela tblProjectAccess, o mesmo campo é utilizado para relacionar o projeto ao seu dono e àqueles que podem acessa-lo.

Por fim as tabelas do módulo Assembly armazenam os dados sobre ORFs e tRNAs encontrados ou genes preditos e também as informações que foram anotadas sobre estes elementos. A vinculação de uma *assembly* a um usuário é realizada por meio do campo idProject na tabela tblAssembly.

O MER pode ser visualizado na Figura 48, onde o conjunto de tabelas de cada módulo está realçado com diferentes cores para melhor visualização e compreensão.

FIGURA 48 – MER



FONTE: O autor (2016).

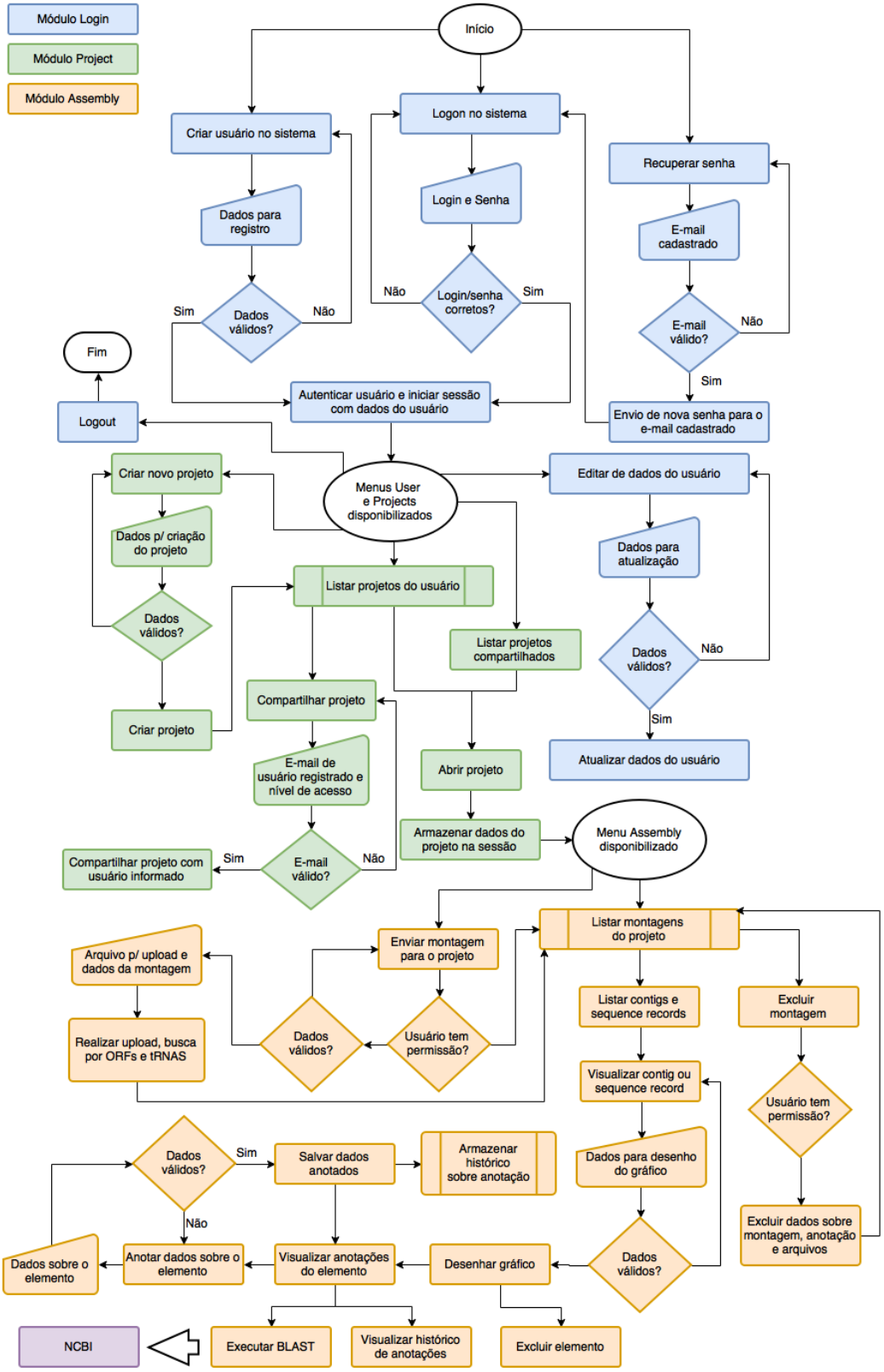
3.5.3 Fluxograma

O fluxo do sistema foi desenhado de acordo com a relação estabelecida entre os módulos e sequência de atividades a serem desempenhadas.

Inicialmente é necessário que um usuário se registre no Web GAAT. A partir do momento em que um usuário está autenticado no sistema, é possível que este crie projetos e os compartilhe com outros usuários. A existência de um projeto, por sua vez, possibilita que arquivos de dados sequenciados sejam enviados e que os recursos do módulo Assembly sejam utilizados.

Na Figura 49 é possível observar o fluxograma, onde o fluxo de cada módulo está realçado com diferentes cores para melhor visualização e compreensão.

FIGURA 49 – FLUXOGRAMA



FONTE: O autor (2016).

4 EXPERIMENTOS PRÁTICOS E RESULTADOS

Ao longo do desenvolvimento deste trabalho, ao analisar diferentes ferramentas para visualização e anotação de genomas, foi evidenciada a viabilidade do desenvolvimento de um ambiente único de trabalho. A abordagem adotada pelo Web GAAT retira do usuário final a necessidade de instalação dos programas integrados à plataforma e facilita a utilização destes por meio de uma interface única, o que demanda menor tempo para familiarização e utilização dos recursos disponibilizados.

Nos tópicos a seguir, a partir de um estudo de caso, serão relatados os resultados dos programas GLIMMER e tRNAscan-SE, integrados à plataforma, bem como as ORFs encontradas pelo script e, principalmente, a visualização destes dados no modelo proposto pelo programa. Também foi realizado outro teste envolvendo criação de usuários e projetos, para validação destes de cenários nos quais podem existir projetos individuais ou colaborativos e diferentes níveis de acesso a diferentes projetos.

4.1 ENVIO DE ARQUIVOS

Para o estudo de caso foi enviado para o Web GAAT um arquivo FASTA de uma sequência hipotética contendo um contig com 5513887 pares de base e 5,6MB de tamanho físico. Conforme resultados discriminados na Tabela 2, foram analisados os seguintes fatores:

- Método: Opção escolhida para busca por ORFs;
- ORFs: Número de ORFs ou genes preditos encontrados com base no método escolhido;
- tRNAs: Números de tRNAs encontrados pelo tRNAscan-SE;
- Upload: Tempo necessário para envio do arquivo para o servidor;
- Processamento: Tempo necessário para execução dos scripts de busca por ORFs e tRNAs;

TABELA 2 - RESULTADOS DOS ARQUIVOS ENVIADOSPARA O CASO DE USO

Método	ORFs	tRNAs	Upload (segundos)	Processamento (minutos)
ATG	34524	55	5,1	12,3
GLIMMER	5215	55	5,2	4,5

FONTE: O autor (2016).

Nota-se que, no primeiro envio, a quantidade de ORFs encontradas é maior. Isto ocorre porque o script realiza a busca baseando-se em critérios textuais, ou seja, qualquer sequência de caracteres (bases) iniciando com ATG e terminando em um códon de término, tendo um número de pares de bases igual ou maior do que o valor definido pelo usuário, é válida. O tempo de processamento também é elevado pois o algoritmo analisa a sequência seis vezes, uma vez para cada *frame* de leitura. No segundo envio os valores obtidos são relativamente menores. Comparando o resultado dos dois processos, é possível considerar o GLIMMER como uma melhor opção uma vez que o programa utiliza o Modelo Oculto de Markov para predição de genes, sendo este mais significante do que uma busca puramente textual, além de demandar menor tempo de processamento para a obtenção dos resultados.

Com relação ao tempo de upload, este pode ser considerado satisfatório, mas pode variar de acordo com tamanho de arquivo e velocidade de conexão com a Internet do usuário.

4.2 VISUALIZAÇÃO E ANOTAÇÃO

A visualização tem um papel fundamental no processo de análise e anotação de genomas. Através dela todas as informações presentes em uma sequência podem ser abstraídas e transformadas em elementos visuais de modo que o usuário possa analisar e interagir com genes com maior facilidade, direcionando seu trabalho a áreas de interesse conforme organização dos dados apresentados.

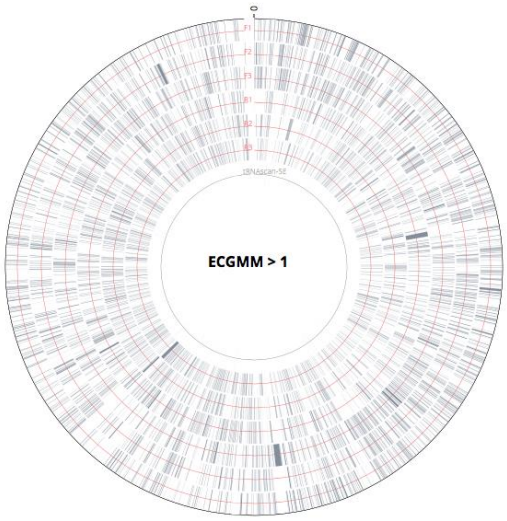
No modelo de visualização proposto pelo Web GAAT, os elementos presentes em uma sequência são distribuídos ordenadamente ao longo de seus respectivos *frames* de leitura, levando em consideração seu início e término, possibilitando assim compreensão de sua posição e tamanho em relação à

porção do *contig* analisado. Para este modelo de visualização foram realizados os seguintes testes:

1. Visualização de todos os elementos presentes no *contig*;
2. Visualização de uma porção do *contig* e aplicação de zoom para visualização de áreas de interesse;
3. Visualização de diferentes elementos (*features*) em meio a sequência;

No primeiro teste, visualização de todos os elementos presentes no *contig*, não foi possível visualizar os resultados obtidos no primeiro com o primeiro arquivo enviado. A grande quantidade de ORFs encontradas, 34524, gerou um alto tempo de resposta do servidor, o qual cancelou a requisição antes da renderização do gráfico. Com relação ao segundo arquivo enviado, o gráfico gerado pelo Web GAAT exibiu todos os 5215 genes preditos pelo GLIMMER, conforme Figura 50.

FIGURA 50 - VISUALIZAÇÃO DE TODOS OS GENES PREDITOS PELO GLIMMER

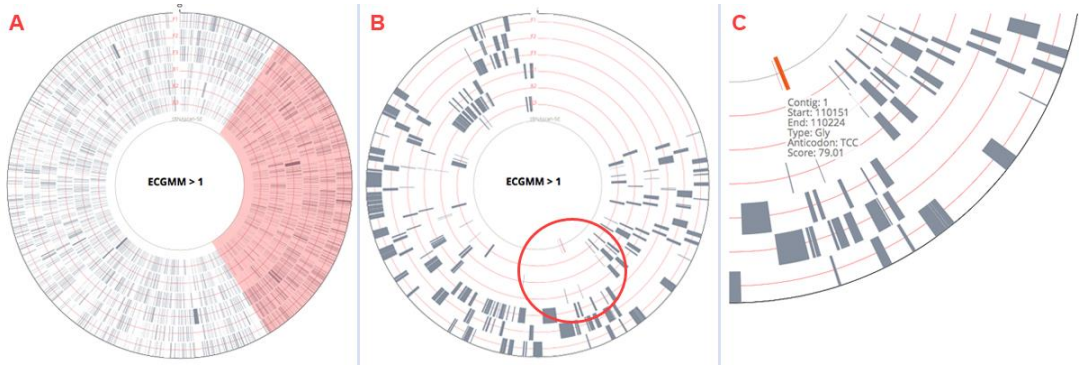


FONTE: O autor (2016).

É possível notar na figura que o modelo atendeu o que foi proposto, possibilitando a visualização e identificação de áreas do genoma com maior concentração de ORFs ou genes preditos e também quais destes elementos se destacam pelo seu tamanho em relação ao *contig*. Este panorama geral serve de base para o segundo teste, que consiste em utilizar os filtros *Start* e *End* para limitar uma porção do genoma para exibição e utilizar o *zoom* para melhor

analisar uma região de interesse. A Figura 51 demonstra a aplicação deste teste com os resultados obtidos com o segundo arquivo enviado.

FIGURA 51 - DELIMITAÇÃO DE UMA ÁREA DE INTERESSE E APLICAÇÃO DE ZOOM



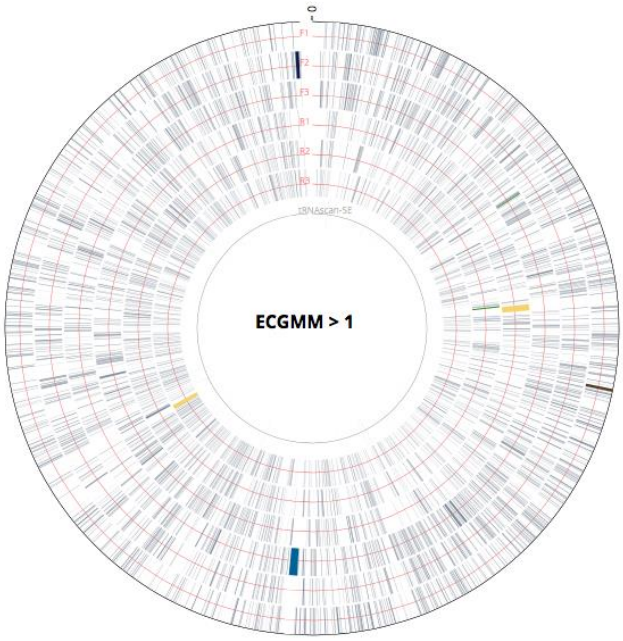
FONTE: O autor (2016).

Na figura, em A, é possível identificar uma área em vermelho que corresponde ao gráfico exibido em B. Nota-se que o Web GAAT redesenhou todos os genes preditos presentes na área marcada em A de modo que o gráfico continue a representar toda uma área determinada pelo usuário. Ainda em B, a área circunscrita corresponde à região onde foi aplicado o zoom para melhor visualização dos elementos, conforme exibido em C, onde o ponteiro do mouse foi posicionado sobre um dos tRNAs encontrados pelo programa tRNAscan-SE.

A mesma metodologia aplicada no segundo teste pode ser utilizada para resolver o problema apresentado ao tentar visualizar todas as ORFs encontradas com o primeiro arquivo enviado. A delimitação de regiões a serem analisadas é, portanto, um recurso válido, especialmente no caso de genomas longos.

O terceiro teste realizado objetivou-se a analisar a caracterização de elementos em meio ao *contig*, para identificação de *features* específicas ao longo da sequência. O resultado se mostrou satisfatório, visto que o contraste de cores possibilita o destaque e diferenciação entre elementos anotados válidos, suas propriedades, e elementos inválidos, conforme Figura 52.

FIGURA 52 - CONTRASTE ENTRE ELEMENTOS ANOTADOS E ELEMENTOS INVÁLIDOS.



FONTE: O autor (2016).

4.3 CRIAÇÃO DE USUÁRIOS E PROJETOS

Para o teste dos módulos Login e Project, desenvolvidos com a finalidade de possibilitar o registro de usuários, projetos e compartilhamento destes a fim de proporcionar o trabalho colaborativo, três usuários foram criados e, para cada um, foi atribuído um projeto, onde os cenários descritos na Tabela 3 foram validados:

TABELA 3 - CENÁRIOS DE TESTE DOS MÓDULOS LOGIN E PROJECT

Cenário	Usuário	Projeto	Cenário
1	A	PA	Usuário A cria e compartilha o projeto PA com B com a permissão *View/Edit* e com C com a permissão *View*.
2	B	PB	Usuário B cria e compartilha projeto PB com A e C atribuindo a estes a permissão *View/Edit.*
3	C	PC	Usuário C cria o projeto PC e compartilha o projeto com A e B com a permissão *View*.
4	D	PD	Usuário D cria projeto PD e não realiza nenhum compartilhamento.

FONTE: O autor (2016).

Em todos os cenários o resultado foi satisfatório, conforme esperado apenas os donos dos projetos puderam realizar o compartilhamento, adicionando ou removendo usuários. Usuários que tinham acesso *View/Edit* aos

projetos PA e PB conseguiram enviar arquivos e realizar anotação sobre os dados. Porém apenas os usuários A e B, criadores dos projetos PA e PB, respectivamente, tinham permissão para excluir sequências enviadas.

Nos projetos PA e PB (cenários 1 e 2), onde mais de um usuário tinha permissão para anotar dados, notou-se que o histórico armazenado pelo Web GAAT é um recurso importante pois este se mostrou muito útil não apenas para registro do usuário responsável pela anotação, mas também para manter um histórico sobre ORFs e genes preditos que foram alterados em diferentes interações.

Por fim, usuários que tinham permissão apenas para visualizar os dados (View), não conseguiram realizar nenhum tipo de anotação e, projetos não compartilhados, não puderam ser acessados por outros usuários.

5 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho propôs a construção de uma plataforma Web que reúne em uma única interface os principais recursos utilizados no processo de análise e anotação de genomas de procariotos. Nomeado Web GAAT: *Genome Analysis and Annotation Tool* (Ferramenta para Análise e Anotação de Genomas), o software utiliza recursos de VI para possibilitar a visualização e interação com dados genômicos sequenciados obtidos a partir do envio de arquivos FASTA e GenBank.

Composto por três módulos, Login, Project e Assembly, a solução proposta foi construída sobre uma arquitetura flexível que possibilita que os recursos apresentados possam ser modificados ou que novas funcionalidades possam ser adicionadas com maior facilidade.

O primeiro módulo, Login, foi criado com o objetivo de possibilitar que usuários possam se registrar e utilizar o programa.

O módulo Project foi direcionado à necessidade de criação de projetos de análise e anotação de genomas, possibilitando que pesquisadores possam trabalhar em projetos individuais ou em projetos compartilhados, onde vários usuários podem acessar dados sequenciados de um mesmo organismo, realizar anotação ou apenas acompanhar o andamento das atividades.

Por fim, o módulo Assembly foi desenvolvido para possibilitar a busca automática por ORFs, tRNAs ou predição de genes a partir do envio de um arquivo FASTA. O módulo retira do usuário a necessidade de utilizar os programas GLIMMER e tRNAscan-SE ao executar os programas automaticamente e possibilitar que o usuário interaja com os resultados em uma interface simples e intuitiva.

O modelo de visualização de dados genômicos sequenciados adotado pelo modulo Assembly, fundamentado em recursos da Visualização da Informação, mostrou-se bastante satisfatório ao possibilitar a análise de regiões completas ou parciais de genomas, permitindo ao usuário identificar dados já anotados e áreas com maior ou menor concentração de elementos.

Outro recurso do módulo Assembly que se mostrou bastante útil é a gravação de histórico sobre dados anotados, o que busca oferecer maior

segurança no trabalho em equipe, pois permite analisar alterações em determinado elemento ao longo de um projeto.

A divisão modular do Web GAAT teve como objetivo facilitar rotinas de manutenção e também possibilitar que novas funcionalidades possam ser adicionadas à plataforma sem a necessidade de grandes alterações, o que aumenta a vida útil do programa. Neste ponto, é possível citar alguns caminhos para desenvolvimento futuro, os quais agregariam valor ao programa. São eles:

- Implementação de um *genome browser* como o Artemis para análise e anotação das ORFs encontradas e genes preditos;
- Implementação de funcionalidades que permitam manipular as sequencias, como por exemplo alterar a posição de início;
- Possibilitar execução local do BLAST, com um banco específico definido pelo usuário;
- Disponibilização de outros recursos, como o RBSFinder (SUZEK et al, 2001), utilizado para encontrar sítios de ligação com ribossomos em genes e genomas bacterianos, usualmente executado a partir de resultados do GLIMMER;

O cumprimento de todos os objetivos propostos pelo trabalho, os quais foram descritos ao longo dos capítulos 3 e 4, resultaram em um software com potencial para utilização no cotidiano de pesquisas relacionadas à análise e anotação de genomas. A constante evolução das tecnologias relacionadas à análise genômica pode trazer novos recursos ao Web GAAT, consolidando o objetivo principal do software, de ser uma plataforma que forneça ferramentas importantes por meio de uma interface única e funcional, possibilitando que usuários possam atuar com maior foco em pesquisas, obtendo melhores resultados.

Em relação ao programa GAAT, acredita-se que seu forte legado, fundamental para o desenvolvimento deste trabalho, pode ser substituído pelo Web GAAT, principalmente à partir da adição de novos recursos que podem enriquecer o que foi produzido.

REFERÊNCIAS

ALTSCHUL SF, GISH W, MILLER W, MYERS EW, LIPMAN DJ. Basic local alignment search tool. **Journal Of Molecular Biology,** v. 215, n. 3, p. 403-410,1990.

ANDREATTA, Anderson. **Bak4bio framework – Brazilian Army knife for bioinformatics.** 82 f. Dissertação (Pós-Graduação em Bioinformática) – Universidade Federal do Paraná, Curitiba, 2013.

ATTWOOD, Teresa; GISEL, Andreas; ERIKSSON, Nils-Einar; BONGCAM-RUDLOFF, Erik. Concepts, Historical Milestones and the Central Place of Bioinformatics in Modern Biology: A European Perspective. **Bioinformatics - Trends and Methodologies.** 2011. Disponível em: <http://www.intechopen.com/books/bioinformatics-trends-and-methodologies/concepts-historical-milestones-and-the-central-place-of-bioinformatics-in-modern-biology-a-european->. Acesso em: 07 jan. 2017.

BAXEVANIS, Andreas D. The Importance of Biological Databases in Biological Discovery. **Current Protocols in Bioinformatics.** 2009. Disponível em: <http://onlinelibrary.wiley.com/doi/10.1002/0471250953.bi0101s27/abstract;jsessionid=7CF734250CFBDFC1A66EDEBE18171C2B.f04t02>. Acesso em: 08 jan. 2017.

CARD, Stuart K; MACKINLAY, Jock D.; SHNEIDERMAN, Ben. **Readings in Information Visualization: Using Vision to Think (Interactive Technologies),** 1. ed. Burlington: Morgan Kaufmann, 1999.

CARDOSO, R.L.A. **Análise Genômica Comparativa de Bactérias do Gênero Herbaspirillum.** 164 f. Tese (Doutorado em Ciências-Bioquímica) – Universidade Federal do Paraná, Curitiba, 2015.

CODEIGNITER. CodeIgniter. Disponível em: <https://www.codeigniter.com/>. Acesso em: 05 jan. 2017.

DIAS, M. P.; CARVALHO, J. O. F. A visualização da informação e a sua contribuição para a ciência da informação. **DataGramaZero**, v. 8, n. 5, p. 0-0, 2007. Disponível em: <http://basessibi.c3sl.ufpr.br/brapci/v/a/4729>. Acesso em: 04 dez. 2016.

EBI. Next Generation Sequencing Practical Course. 2012. Disponível em: <http://www.ebi.ac.uk/training/online/course/ebi-next-generation-sequencing-practical-course>. Acesso em: 05 jun 2016.

FREITAS, C. M. D. S.; CHUBACHI, O. M.; LUZZARDI, P. R. G. ; CAVA, R. A. Introdução à Visualização de Informações. **RITA**, v. VII, n.2, 2001.

GAMMA, Erich; VLISSIDES, John; JOHNSON, Ralph; HELM, Richard. **Design Patterns: Elements of Reusable Object-Oriented Software,** 1. ed. Addison-Wesley, 1995.

GMOD. Web Apollo. Disponível em: <http://gmod.org/wiki/WebApollo>. Acesso em 18 dez. 2016.

HARVARD. Model, View, Controller (MVC) Design Pattern. 2010. Disponível em: <http://cscie153.dce.harvard.edu/>. Acesso em: 11 set 2016.

KOONIN EV, GALPERIN MY. **Sequence - Evolution - Function: Computational Approaches in Comparative Genomics**. 1. ed. Boston: Kluwer Academic, 2003.

KREMER, Frederico Schmitt; PINTO, Luciano da Silva. Anotação de Genomas. 2016. Disponível em: <http://labbioinfo.ufpel.edu.br/aulas/Cap%EDtulo%205.pdf>. Acesso em 30 nov 2016.

LEE, Eduardo; HELT, Gregg A; REESE, Justin T; MUNOZ-TORRES, Monica C; CHILDERS, Chris P; BUELS, Robert M; STEIN, Lincoln; HOLMES, Ian H; ELSIK,Christiane G; LEWIS, Suzana E. Web Apollo: a web-based genomic annotation editing platform. **Genome Biology,** 14: r93, 2013.

LEFF, Avraham; RAYFIELD, James T. Web-Application Development Using the Model/View/Controller Design Pattern. **IEEE Enterprise Distributed Object Computing Conference,** p. 118-127, 2001.

LOWE, TM; EDDY, SR. tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. **Nucleic Acids Research,** v. 25, n. 5, p. 955–964, 1997.

LIMA, Manuel. **Visual Complexity: Mapping Patterns of Information.** 1. ed. Nova Iorque: Princeton Architectural Press, 2011.

LOCAWEB. LocawebStyle. Disponível em: <http://opensource.locaweb.com.br/locawebstyle/>. Acesso em: 05 jan. 2017.

MANGALAN, Harry; ROBINSON, James T; MESIROV, Jill P. Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. **Briefings in Bioinformatics,** Oxford, v. 3, n.3, p. 296-302, 2002.

MICROSOFT. SQL Server. Disponível em: < https://www.microsoft.com/pt-br/sql-server/>. Acesso em: 02 nov. 2016.

MIRANDA, Willian. Os 5 Bancos de Dados mais utilizados do Mercado. Disponível em: <http://aprendaplsql.com/oracle/os-5-bancos-de-dados-mais-utilizados-do-mercado/>. Acesso em: 02 nov. 2016.

MYSQL. Disponível em: <https://www.mysql.com/>. Acesso em: 02 nov. 2016.

MONGODB. Disponível em: <https://www.mongodb.com/>. Acesso em: 02 nov. 2016.

NATIONS, Daniel. What is a Web Application. Disponível em: <http://webtrends.about.com/od/webapplications/a/web_application.htm>. Acesso em: 22 out 2016.

NASCIMENTO, Hugo A. D.; FERREIRA, Cristiane B. R. Uma Introdução Á à Visualização de Informações. **Visualidades**, v. 9, n. 2, p. 13-43, 2011. Disponível em: <https://www.revistas.ufg.br/VISUAL/article/viewFile/19844/12233>. Acesso em: 04 dez. 2016.

NCBI. BLAST. Disponível em: <http://blast.ncbi.nlm.nih.gov/Blast.cgi >. Acesso em: 22 out 2016.

NCBI. Open Reading Frame. Disponível em: <https://www.ncbi.nlm.nih.gov/books/NBK5191/#IX-O>. Acesso em: 20 out 2016.

NCBI. ORF Finder. Disponível em: <http://www.ncbi.nlm.nih.gov/orffinder/>. Acesso em: 22 out 2016.

OMICTOOLS. BioCircos. Disponível em: <https://omictools.com/biocircos-js-tool>. Acesso em: 08 jan. 2017.

ORACLE. Oracle Database. Disponível em: <https://www.oracle.com/br/index.html>. Acesso em: 02 nov. 2016.

PETERSEN, Jeremy. Benefits of using the n-tiered approach for web applications. 2008. Disponível em: <http://www.adobe.com/devnet/coldfusion/articles/ntier.html> . Acesso em: 14 ago 2016.

PISA, F. R. O. Desenvolvimento de programa integrado de montagem e anotação de genomas. Relatório do Programa de Iniciação Cientifica. Curitiba: Departamento de Bioquímica e Biologia Molecular, Bioinformática do Projeto GENOPAR/2004014420; Ago 2008.

RIEHLE, Dirk. Framework Design: A Role Modeling Approach. 2000. Disponível em: <http://dirkriehle.com/computer-science/research/dissertation/diss-a4.pdf>. Acesso em: 14 ago 2016.

RUSSEL, Jesse; COHN, Ronald. **Web Development,** Impressão sob demanda. Book On Demand, 25 de janeiro de 2013.

RUTHERFORD, Kim; PARKHILL, Julian; CROOK, James; HORSNELL, Terry; RICE, Peter; RAJANDREAM, Marie-Adèle; BARREL, Bart. Artemis: sequence visualization and annotation. **Bioinformatics Applications Note,** Oxford, v. 16, n.10, p. 944-945, 2000.

SANGER. Artemis. 2014. Disponível em: http://www.sanger.ac.uk/science/tools/artemis. Acesso em: 22 out 2016.

SALZBERG, Steven L; DELCHER, Arthur L; KASIF, Simon; WHITE. Microbial gene identification using interpolated Markov models. **Nucleic Acids Research,** Oxford, v. 26, n.2, p. 544-548, 1998.

SILBERSCHATZ, Abraham; KORTH, Henry F; SUDARSHAN, S. **Sistema de Banco de Dados.** 5. ed. Rio de Janeiro: Campus Elsevier, 2006.

SKINNER, E.; MITCHELL; UZILOV; ANDREW; STEIN, D.; LINCOLN; MUNGALL, Christopher; HOLMES, H.; Ian. JBrowse: A next-generation genome browser. **Genome research.** 19. 1630-8. 10.1101/gr.094607.109., 2009.

STEIN Lincoln. Genome annotation: from sequence to biology. Nature Reviews Genetics, London, n.2 p. 493-505, 2001.

SUZEK, Baris E; ERMOLAEVA, Maria D; SCHREIBER, Mark; SALZBERG, Steven L. A probabilistic method for identifying start codons in bacterial genomes. **Bioinformatics,** Oxford, v. 17, n.12, p. 1123-1130, 2001.

TAO, Ying; LIU, Yang; FRIEDMAN, Carol; LUSSIER, Yves. Information Visualization Techniques in Bioinformatics during the Postgenomic Era. **Drug Discovery Today Biosilico,** Washington, v.2 n.6, p.237-245, 2004.

TECHOPEDIA. Modular Programming. 2018. Disponível em <https://www.techopedia.com/definition/25972/modular-programming>. Acesso em: 15/04/2018.

THATAVARTHI, Pravallika; SURESH, Betam. An Application to prevent SQL Injection Attacks using Randomized Encryption Algorithm. **International Journal of Computer Trends and Technology**, v.4, n.8, p. 178-192, 2013.

THORVALDSDÓTTIT, Helga; ROBINSON, James T; MESIROV, Jill P. Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. **Briefings in Bioinformatics,** Oxford, v. 14, n.2, p. 178-192, 2013.

TIEPPO, EDUARDO. **Montagem e Análise Preliminar do Genoma de Bradyrhizobium elkanii 587 Utilizando Leituras de Sequências de DNA Curtas.** 77 f. Dissertação (Mestrado em Bioinformática) – Universidade Federal do Paraná, Curitiba, 2011.

VIRTUALBOX. VirtualBox. Disponível em: <https://www.virtualbox.org/>. Acesso em: 31/01/2017.

YANDELL, Mark; ENCE, Daniel. A beginner's guide to eukaryotic genome annotation. **Nature Reviews Genetics,** London, n.13 p. 329-342, 2012.

ZAHA, Arnaldo; FERREIRA, Henrique B; PASSAGLIA, Luciane M.P. **Biologia Molecular Básica,** 5. ed. Porto Alegre: Artmed, 2014.