



**UNIVERSIDADE DO ESTADO DA BAHIA**  
**DEPARTAMENTO DE CIÊNCIAS EXATAS E DA TERRA**  
**CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO**

**GABRIEL ANTÔNIO ALVES DE AMORIM**

**FUNREGULATION: CONCEPÇÃO DE UMA ARQUITETURA WEB (BACK-END)**  
**QUE IMPLEMENTE UM PIPELINE PARA CONSTRUÇÃO DE REDES**  
**REGULATÓRIAS DE GENES DE FUNGOS**

**SALVADOR**

**2023**

GABRIEL ANTÔNIO ALVES DE AMORIM

FUNREGULATION: CONCEPÇÃO DE UMA ARQUITETURA WEB (BACK-END) QUE  
IMPLEMENTE UM PIPELINE PARA CONSTRUÇÃO DE REDES REGULATÓRIAS DE  
GENES DE FUNGOS

Monografia apresentada ao curso de Sistemas de Informação do Departamento de Ciências Exatas e da Terra da Universidade do Estado da Bahia - UNEB, como requisito à obtenção do grau de Bacharel em Sistemas de Informação. Área de Concentração: Ciência da Computação

Orientador: Alexandre Rafael Lenz

SALVADOR

2023

GABRIEL ANTÔNIO ALVES DE AMORIM

FUNREGULATION: CONCEPÇÃO DE UMA ARQUITETURA WEB (BACK-END) QUE  
IMPLEMENTE UM PIPELINE PARA CONSTRUÇÃO DE REDES REGULATÓRIAS DE  
GENES DE FUNGOS

Monografia apresentada ao curso de Sistemas de Informação do Departamento de Ciências Exatas e da Terra da Universidade do Estado da Bahia - UNEB, como requisito à obtenção do grau de Bacharel em Sistemas de Informação. Área de Concentração: Ciência da Computação

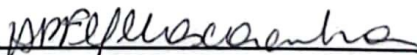
Aprovada em:

BANCA EXAMINADORA



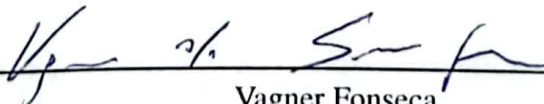
---

Alexandre Rafael Lenz (Orientador)  
Universidade do Estado da Bahia – UNEB



---

Ana Patrícia Fontes Magalhães Mascarenhas  
Universidade do Estado da Bahia – UNEB



---

Vagner Fonseca  
Universidade do Estado da Bahia – UNEB

## Termo de Anuência do Orientador

Declaro para os devidos fins que li e revisei este trabalho e atesto sua qualidade como resultado final desta monografia. Confirmando que o referencial teórico apresentado é completo e suficiente para fundamentar os objetivos propostos e que a metodologia científica utilizada e os resultados finais são consistentes e com qualidade suficiente para submissão à banca examinadora final do Trabalho de Conclusão de Curso II do curso de Bacharelado em Sistemas de Informação.



---

Prof. Alexandre Rafael Lenz



## **AGRADECIMENTOS**

A Deus, pela minha vida, e por me permitir ultrapassar todos os obstáculos encontrados ao longo da realização deste trabalho. Minha família e amigos por todo o apoio ao longo do curso. A todos os professores que contribuíram para minha formação profissional e pessoal. Ao meu orientador, Alexandre Rafael Lenz, pela paciência e carinho durante toda construção deste trabalho. A Eduardo Balbinot, pela contribuição no processo de construção do software.

"A maioria das pessoas está tão absorta na contemplação do mundo exterior que está totalmente alheia ao que está acontecendo em si"

(Nikola Tesla)

## RESUMO

A construção de Redes Regulatórias de Genes, do inglês, *Gene Regulatory Network* (GRN) é um tema importante na área de bioinformática pois, a partir do entendimento de uma rede de genes, é possível realizar uma série de estudos para o desenvolvimento de diferentes áreas como saúde, agricultura e industrial. Contudo a compreensão das relações entre os genes de um ser vivo é um grande desafio tanto na área da biologia experimental quanto na área computacional. O objetivo desta pesquisa foi desenvolver uma arquitetura Web (*Back-end*) que fosse capaz de facilitar a construção de GRNs de fungos, a partir de uma arquitetura de *software* responsável por unificar ferramentas já validadas e utilizadas na bioinformática. Primeiramente foi necessária a identificação das etapas e ferramentas que iriam compor o *pipeline*, responsável pela construção da GRN. Em seguida foi preciso definir um modelo para realizar o gerenciamento e o escalonamento, das etapas e ferramentas responsáveis por construir a GRN. Por fim tínhamos como objetivos específicos a definição de um modelo que fosse capaz de encapsular as ferramentas de terceiros como ProteinOrtho e RSAT, e a construção de duas GRNs de fungos de ordens distintas, para a validação do trabalho. Na validação deste trabalho é mostrado a construção de duas GRNs de dois fungos, o primeiro fungo é o *Madurella mycetomatis* causador de uma doença chamada Micetoma, comum na África, América do Sul e Índia. O segundo é o *Cordyceps militaris* bastante utilizado na medicina chinesa há séculos por atuar na melhora do sistema imunológico, equilíbrio hormonal e diversos outros benefícios à saúde. Determinamos que a arquitetura web desenvolvida (*Back-end*) possui atributos dignos de nota que merecem ser mencionados. O primeiro aspecto fundamental é a capacidade da arquitetura de receber melhorias e modificações futuras. Por exemplo, todas as tarefas dentro da arquitetura operam de forma assíncrona através de uma solução modular. O segundo aspecto crucial é a sua contribuição para a compreensão da relação entre os genes dos fungos. A arquitetura desenvolvida nesta pesquisa tem potencial para unificar todo o processo e ferramentas de geração de GRNs. Portanto, ao integrar esta arquitetura com o *Front-end*, a comunidade científica que estuda fungos ou GRNs pode utilizar este *software* para avançar em suas pesquisas científicas.

**Palavras-chave:** Bioinformática. Redes regulatórias de genes. fungos;

## ABSTRACT

The construction of Gene Regulatory Networks (GRNs) is an important subject in bioinformatics because, with the understanding of a gene network, it is possible to make studies on the development of different areas, like health, agriculture, and industry. However, comprehending the relations between the genes of a living being is a big challenge both in biological experimentation and computational areas. This research aims to develop a web architecture (back-end) that will facilitate the construction of fungi's GRN from a software architecture responsible for unifying tools already validated and used in bioinformatics. Firstly, it was necessary to identify the phases and tools that would compose the pipeline responsible for generating the GRN. After that, it was necessary to define a model to manage and escalate the phases and provide the tools responsible for creating the GRN. Lastly, we had as specific goals the definition of a model that would be responsible for encapsulating third party tools, like ProteinOrtho and RSAT, and the construction of two GRNs of different fungi to validate the research. The conclusion of this research shows the construction of GRN of two fungi. The first fungus is *Madurella mycetomatis*, which causes a disease called micetoma, which is very common in Africa, South America, and India. The second one is *Cordyceps militaris*, widely used in Chinese medicine for centuries, acting on the improvement of the immunological system, hormonal balance and several other benefits to health. We have determined that the developed web architecture (back-end) possesses noteworthy attributes that warrant mentioning. The first key aspect is the architecture's capacity for receiving future improvements and modifications. For instance, all the tasks within the architecture operate asynchronously through a modular solution. The second crucial aspect is its contribution to understanding the relationship among Fungus genes. The architecture developed in this research has the potential to unify the entire process and tools for generating GRNs. Therefore, upon integrating this architecture with the front end, the scientific community studying fungi or GRNs can utilise this software to advance their scientific research.

**Keywords:** Bioinformatics. Gene Regulatory Network. Fungi

## LISTA DE FIGURAS

Figura 1 – Áreas onde os conhecimentos sobre fungos podem ser utilizados. . . . .	16
Figura 2 – Exemplo de uma GRN. . . . .	17
Figura 3 – Fontes de dados para construção de GRNs. . . . .	18
Figura 4 – Conceito de ortologia . . . . .	19
Figura 5 – Como é construída uma PWM . . . . .	19
Figura 6 – Funcionamento da ferramenta ProteinOrtho . . . . .	22
Figura 7 – Funcionamento de um sistema de filas com Celery e RabbitMQ . . . . .	26
Figura 8 – Lista com os trabalhos correlatos . . . . .	28
Figura 9 – Adaptação da <i>Design Science Research</i> para esta Pesquisa . . . . .	30
Figura 10 – Ciclo de desenvolvimento do <i>FunRegulation</i> . . . . .	32
Figura 11 – Fluxograma com os passos da construção de uma GRN . . . . .	37
Figura 12 – Arquitetura ( <i>Back-end</i> ) do Sistema FunRegulation . . . . .	41
Figura 13 – Rotas de Execução do Pipeline de dois usuários distintos . . . . .	42
Figura 14 – Retorno do <i>end-point</i> para listar organismos . . . . .	43
Figura 15 – Exemplo de requisição para criação de GRN . . . . .	44
Figura 16 – Exemplo de retorno da requisição para criação da GRN . . . . .	44
Figura 17 – Exemplo de retorno de consulta de estado da tarefa . . . . .	45
Figura 18 – Exemplo de retorno com informações de um nó da GRN . . . . .	46
Figura 19 – Exemplo de retorno com medidas de centralidade um nó da GRN . . . . .	46
Figura 20 – Exemplo de requisição para criação de um usuário . . . . .	47
Figura 21 – E-mail para confirmação da conta do usuário . . . . .	47
Figura 22 – E-mail com link para efetuar troca de senha . . . . .	48
Figura 23 – GRN do fungo <i>Madurella mycetomatis</i> gerada pela FunRegulation . . . . .	50
Figura 24 – GRN ampliada do fungo <i>Madurella mycetomatis</i> . . . . .	51
Figura 25 – Principais estatísticas da rede do fungo <i>Madurella mycetomatis</i> . . . . .	51
Figura 26 – GRN do fungo <i>Cordyceps militaris</i> gerada pela FunRegulation . . . . .	52
Figura 27 – Principais estatísticas da rede do fungo <i>Cordyceps militaris</i> . . . . .	53

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>15</b>
<b>2.1</b>	<b>Área de Aplicação</b>	<b>15</b>
2.1.1	Fungos	15
2.1.2	Redes Regulatórias de Genes (GRNs)	15
2.1.3	Fontes de Dados para Construção de GRNs	17
2.1.3.1	Ortologia	18
2.1.3.2	Sítios de Ligação de Fatores de Transcrição (TFBSs)	19
<b>2.2</b>	<b>Área Computacional</b>	<b>20</b>
2.2.1	Pipeline para Construção de GRNs	20
2.2.1.1	NCBI - GenBank®	21
2.2.1.2	ProteinOrtho	21
2.2.1.3	CIS-BP	22
2.2.1.4	RSAT ( <i>Regulatory Sequence Analysis Tools</i> ) - <i>matrix-scan</i>	22
2.2.1.5	<i>Python</i>	23
2.2.1.6	PostgreSQL	24
2.2.2	Gerenciamento e Escalonamento de Tarefas	24
2.2.2.1	<i>Celery</i>	25
2.2.2.2	RabbitMQ	25
<b>2.3</b>	<b>Trabalhos correlatos</b>	<b>26</b>
2.3.1	Seleção de trabalhos correlatos	26
2.3.2	Trabalhos selecionados	27
<b>3</b>	<b>METODOLOGIA</b>	<b>29</b>
<b>3.1</b>	<b>Metodologia de Desenvolvimento</b>	<b>31</b>
<b>4</b>	<b>FUNREGULATION</b>	<b>33</b>
<b>4.1</b>	<b>O sistema FunRegulation</b>	<b>33</b>
4.1.1	Levantamento de Requisitos	35
4.1.2	Análise de requisitos	35

<b>4.2</b>	<b>Identificação das Etapas e Ferramentas para Composição do <i>Pipeline</i></b> . . .	<b>36</b>
4.2.1	Instalação das ferramentas . . . . .	37
4.2.2	O funcionamento das etapas e ferramentas que compõem o <i>pipeline</i> . . . . .	38
<b>4.3</b>	<b>Desenho da Arquitetura</b> . . . . .	<b>40</b>
4.3.1	Implementação do Gerenciamento e Escalonamento de Tarefas . . . . .	42
4.3.1.1	Lógica de Encadeamento das <i>tasks</i> . . . . .	42
<b>4.4</b>	<b>Implementação dos requisitos - <i>End-points</i></b> . . . . .	<b>43</b>
4.4.1	Listar organismos - RF07 . . . . .	43
4.4.2	Criar uma GRN - RF06 . . . . .	44
4.4.3	Acompanhar as tarefas do pipeline - RF06 . . . . .	45
4.4.4	Obter informações de um nó da rede - RF10 . . . . .	45
4.4.5	Obter medidas de centralidade de um nó da GRN - RF11 . . . . .	45
4.4.6	Criação de novo usuário - RF01 e RF02 . . . . .	46
4.4.7	Login - RF03 . . . . .	47
4.4.8	Logout - RF04 . . . . .	48
4.4.9	Solicitação de alteração de senha - RF05 . . . . .	48
4.4.10	Confirmação de alteração de senha - RF05 . . . . .	48
<b>5</b>	<b>VALIDAÇÃO</b> . . . . .	<b>50</b>
<b>6</b>	<b>CONSIDERAÇÕES FINAIS</b> . . . . .	<b>54</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>56</b>
	<b>APÊNDICES</b> . . . . .	<b>61</b>
	APÊNDICE A – Revisão sistemática . . . . .	<b>62</b>
	<b>ANEXOS</b> . . . . .	<b>80</b>
	ANEXO A – Documento de Definição de Requisitos . . . . .	<b>81</b>
	ANEXO B – Documento de casos de uso . . . . .	<b>105</b>

# 1 INTRODUÇÃO

A bioinformática é uma área multidisciplinar, definida como a aplicação de ferramentas de computação e análise para capturar e interpretar dados biológicos, sendo esta uma área essencial para manipulação de dados na medicina e biologia moderna (1). Dados biológicos são de extrema importância para compreender os processos biológicos de qualquer ser vivo presente no planeta, como plantas, fungos, animais, vírus (2).

A compreensão dos processos biológicos presentes nos diferentes seres vivos é uma tarefa comumente complexa e que demanda muitos recursos computacionais e humanos. Para iniciar o estudo de um ser vivo é necessário que se faça, primeiramente, o sequenciamento e anotação de seu genoma.

A partir de um genoma completo é possível realizar diversos estudos e análises como, por exemplo, análises genômicas visando compreender o funcionamento de um determinado processo biológico, a causa de uma doença, etc. Análises genômicas são tarefas complexas e requerem um grande esforço por parte dos cientistas. A complexidade dessas tarefas ocorre principalmente devido ao maciço volume de dados, com diferentes formatos e obtidos de várias fontes distintas, dificultando a busca por informações relevantes de maneira rápida e exigindo o uso de uma grande gama de *softwares* distintos (3).

A construção de redes regulatórias de genes, ou em inglês, *Gene Regulatory Networks* (GRN) auxilia a compreensão das relações entre os genes de um ser vivo e, correspondem a dados organizados em forma de grafos que conectam genes reguladores e regulados (4). Os genes reguladores são também chamadas de *Transcript Factors* (TF's) e podem atuar ativando ou reprimindo a expressão de um gene alvo, enquanto os genes alvos (em inglês, *Target Genes* (TG's)) são os genes regulados (5). Compreender essas redes é extremamente importante, pois nos ajudam a entender as relações entre os genes de um ser vivo. Porém, construir essas redes também é bastante desafiador no âmbito da biologia experimental e computacional (2).

O advento das tecnologias de sequenciamento de nova geração possibilitou avanços significativos no campo do conhecimento dos genomas dos organismos. Com a redução dos custos de sequenciamento e aumento na rapidez do processamento das amostras, o número de

genomas sequenciados e disponíveis em bancos de dados cresceu exponencialmente (6). O tamanho e a quantidade de genes dos genomas variam bastante entre as espécies de fungos. Geralmente, os genomas dos fungos filamentosos contêm muitos genes codificadores de proteínas, podendo variar entre 9.000 (nove mil) e 17.000 (dezessete mil) genes (7). A construção de GRNs deve levar em consideração essa demanda por ferramentas robustas, que sejam capazes de manipular rapidamente grandes conjuntos de genes de diversos organismos.

Atualmente existem poucas ferramentas capazes de construir uma GRN para fungos e a maioria possui diversas limitações. Conforme visto na revisão sistemática, no apêndice A, algumas apenas permitem a construção de GRNs para uma determinada espécie ou gênero, enquanto outras ferramentas são limitadas a alguns poucos TFs e TGs. Por exemplo, a ferramenta *MALBoost* permite apenas a construção da GRN para *Plasmodium falciparum*, organismo causador da malária (8).

As GRNs são fundamentais para a compreensão de como genes e processos biológicos estão inter-relacionados em um organismo. Conforme visto, a GRN é uma base extremamente importante para auxiliar estudos experimentais a fim de compreender a rede genômica e possibilitar o entendimento das funções dos genes e seus respectivos processos biológicos (2). Essas redes regulatórias auxiliam no processo de tomada de decisão, como por exemplo, escolhas de genes alvo para engenharia genética, identificação de rotas de regulação, identificação de agrupamentos de genes relacionados a processos biológicos, etc (2). Por consequência, o conhecimento sobre regulação gênica de fungos, por exemplo, pode ser aplicado à produção de biocombustíveis, prospecção de novos medicamentos, produção de alimentos e bebidas, entre outras inúmeras aplicações.

Este trabalho realizou uma revisão sistemática de literatura com o objetivo de obter o conhecimento vigente sobre a construção de GRNs, e as principais ferramentas que são utilizadas no desenvolvimento de *softwares* para este fim. Também buscou elencar os principais *frameworks* utilizados pelas ferramentas para lidar com filas de tarefas e chamadas de *softwares* de terceiros. Por fim, a revisão permitiu avaliar em que medida a questão problema levantada era coerente com um trabalho de pesquisa no nível adequado a um TCC, e avaliar a relevância da proposta no seu conjunto, tendo em vista que uma pesquisa voltada para a construção de uma plataforma Web para construir GRNs de fungos já poderia ter sido desenvolvida.

Portanto, com base no percurso inicial de pesquisa descrito acima, estabelecemos

a seguinte questão norteadora a ser respondida nas etapas seguintes desta investigação: Como facilitar o processo de construção de GRNs de fungos pela integração de ferramentas tecnológicas existentes da bioinformática a partir do conceito de *pipeline*?

Outro aspecto relevante constatado durante a revisão sistemática foi que existem poucos *softwares* que abordam a construção de GRNs e com disponibilidade via Web. Não foram encontrados *softwares* capazes de construir GRNs para fungos. Indicando assim a viabilidade desta pesquisa que pode enriquecer a bioinformática e a micologia.

O FunRegulation é uma ferramenta *desktop*, que foi desenvolvida em um trabalho de doutorado por Alexandre Rafael Lenz. Esta ferramenta foi publicada em dois artigos (5, 9), e está disponível no GitHub (10). Atualmente o FunRegulation está em fase de evolução para Web, sendo dividido em dois trabalhos de conclusão de curso, compreendendo a construção do *Back-end* e do *Front-end*.

Este trabalho então propõe a construção de uma arquitetura Web *Back-end*, capaz de construir GRNs. Uma arquitetura Web que implemente um *pipeline* para construir GRNs de fungos deve ser capaz de unificar etapas que são muito onerosas para serem executadas manualmente. Um sistema Web seria um facilitador de todo o processo de construção de GRNs. O encapsulamento das etapas que compõem o *pipeline* (coleta de dados genômicos, identificação de genes ortólogos e predição de sítios de ligação de fatores de transcrição) agiliza e torna mais eficiente a construção de GRNs, permitindo que usuários não precisem se preocupar com a instalação e execução de diversas ferramentas e nem com a manipulação e armazenamento de arquivos intermediários.

Entendemos que uma ferramenta Web que automatiza a construção de GRNs, a partir de um *pipeline* que une as diversas ferramentas necessárias ao processo, permitirá que a comunidade científica construa GRNs de diversos outros fungos. Por consequência, representando um passo importante para a ciência, no que diz respeito ao conhecimento de regulação de genes de fungos.

Partindo das lacunas e justificativas descritas, os objetivos desta pesquisa são apresentados a seguir:

O objetivo geral deste trabalho propõe a concepção de uma arquitetura Web (*Back-end*) que implemente um *pipeline* para integrar as etapas de construção de redes regulatórias de

genes de fungos.

Para atingir o objetivo geral, os seguintes objetivos específicos foram elaborados:

- Identificar as etapas e ferramentas necessárias para compor o *pipeline* do processo de construção de GRNs.
- Definir um modelo para gerenciar e escalonar as etapas do *pipeline* de construção de GRNs.
- Definir um modelo responsável por encapsular ferramentas de terceiros.
- Construir a GRN de dois fungos de ordens diferentes, para validar a proposta e identificar suas potencialidades e fragilidades.

Os capítulos subsequentes abordam o referencial teórico (Capítulo 2), explicando sobre pontos importantes da área da bioinformática, da área computacional e são demonstrados alguns trabalhos correlatos. No capítulo 3 (Metodologia) é explicada a metodologia de pesquisa e desenvolvimento, utilizadas para a realização deste trabalho. No capítulo 4 (*FunRegulation*) é explicada a arquitetura e as ferramentas, que compõem o *FunRegulation* e o funcionamento de *End-points* importantes do sistema, juntamente com os requisitos do *FunRegulation*. No capítulo 5 (Validação) mostramos como foi realizada a validação deste trabalho e por fim, no capítulo 6 (Considerações finais), são explicados as potencialidades e fragilidades deste trabalho, e como possíveis trabalhos futuros podem aperfeiçoar este trabalho.

## 2 REFERENCIAL TEÓRICO

Para o desenvolvimento desta pesquisa, considerando todos os seus aspectos conceituais e de ordem prática, é fundamental um embasamento teórico. Este capítulo introduz os conceitos relacionados ao trabalho proposto. Estes conceitos estão organizados em três seções: área de aplicação, área computacional e trabalhos correlatos.

### 2.1 ÁREA DE APLICAÇÃO

Nesta seção são apresentados os principais conceitos da área de aplicação desta pesquisa. São citados trabalhos que, em sua maioria, são dissertações de mestrado, teses de doutorado ou artigos científicos e servem como base, não só para o desenvolvimento da pesquisa em si, mas também para o domínio de conceitos fundamentais sobre fungos, GRNs, *pipelines*, entre outros.

#### 2.1.1 Fungos

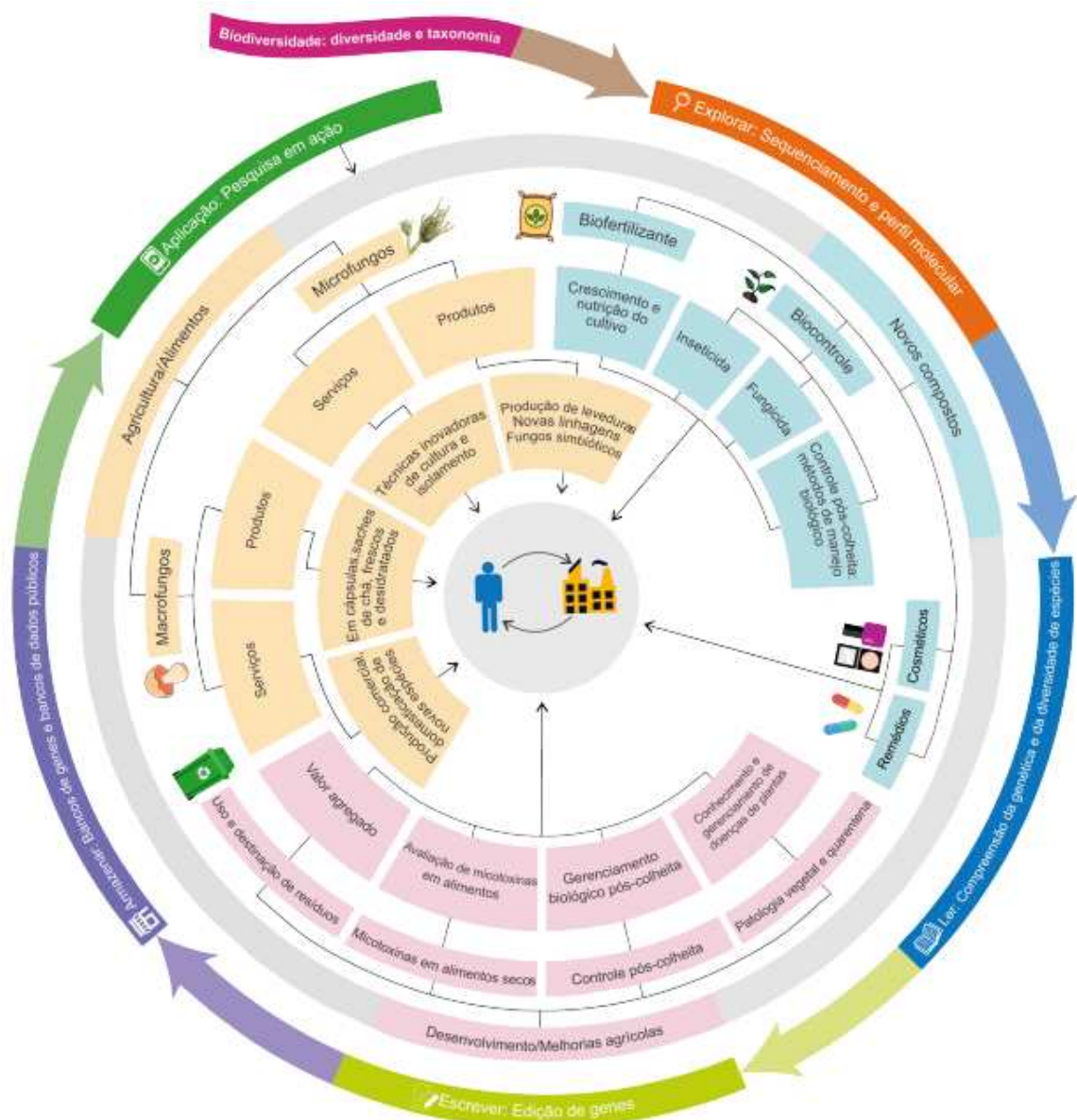
Com 150 mil espécies de fungos já descritas pela ciência, a biodiversidade do reino *Fungi* ainda é pouco explorada (11). O potencial dos fungos pode ser mensurado pelas possibilidades de desenvolvimento de medicamentos, cosméticos, fertilizantes e inseticidas, produção de alimentos, produção de combustíveis, entre outros, conforme demonstrado na Figura 1.

Porém, apesar de todo esse potencial desses seres vivos, pouquíssimos estudos sobre fungos são realizados, muito por conta dos poucos recursos e financiamentos destinados para a realização destas pesquisas (5).

#### 2.1.2 Redes Regulatórias de Genes (GRNs)

Um dos principais desafios da biologia tanto no campo experimental, quanto no campo computacional, é a compreensão das relações entre os genes e os produtos que eles codificam. A identificação de relações regulatórias entre reguladores da transcrição e seus alvos é essencial para a compreensão de fenômenos biológicos que variam do crescimento e divisão

Figura 1 – Áreas onde os conhecimentos sobre fungos podem ser utilizados.



Fonte: Traduzido de (12)

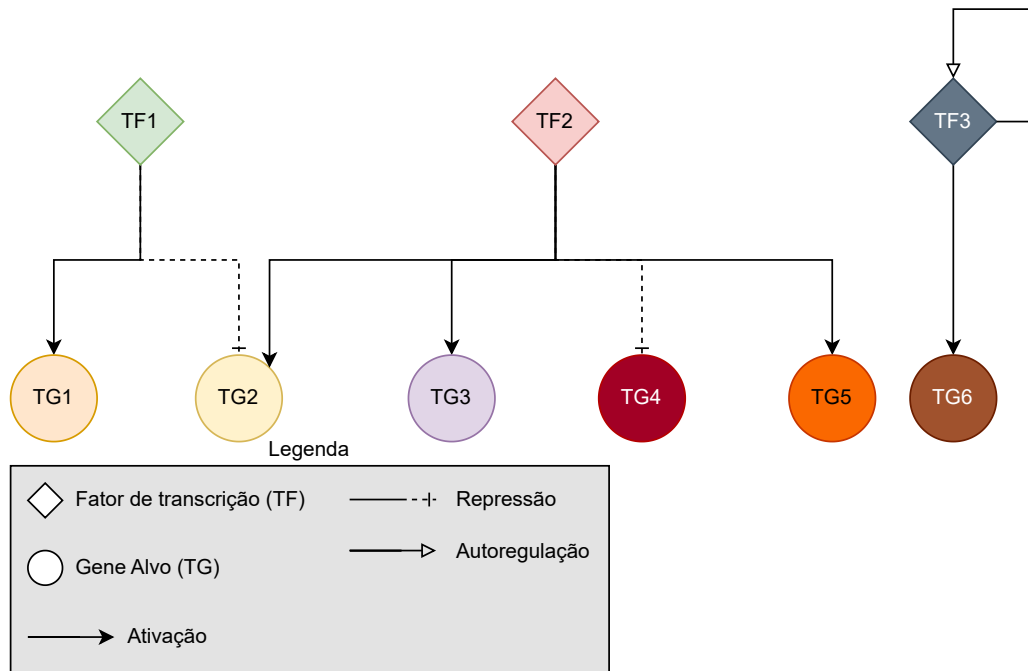
celular à diferenciação e desenvolvimento celular (2).

Conforme já citado na seção introdutória, uma das maneiras de se compreender essas relações entre genes, consiste na construção das Redes Regulatórias de Genes (GRNs). GRNs são constituídas por dados organizados em forma de grafos que conectam genes reguladores aos genes regulados (4). Os genes reguladores são conhecidos como fatores de transcrição (*Transcript Factors* - TFs), já os genes regulados também são conhecidos como genes alvo (*Target Genes* - TGs) (2).

Compreender essas interações é extremamente importante para entender processos biológicos, fenótipos e doenças, (5). Um exemplo de GRN pode ser observado na Figura 2.

Neste exemplo de GRN bastante simplificado, podemos observar que o TF1 regula a expressão de dois TGs (TG1 e TG2). A relação entre TF1 e TG1 é de ativação da expressão de TG1, enquanto que na relação entre TF1 e TG2 ocorre justamente o inverso, ou seja, ocorre a repressão da expressão de TG2. Também é possível observar a possibilidade de autorregulação de um TF, como no caso do TF3.

Figura 2 – Exemplo de uma GRN.



Fonte: O autor

### 2.1.3 Fontes de Dados para Construção de GRNs

A construção de uma GRN pode utilizar distintas fontes de dados como ortologia, sítios de ligação de fatores de transcrição (do inglês *Transcription Factor Binding Sites* (TFBSs)), dados de coexpressão gênica, entre outros conforme exibido na Figura 3.

Neste trabalho utilizamos duas fontes: i) a ortologia, que visa comparar as proteínas do fungo selecionado pelo usuário para construção da GRN com as proteínas de um fungo modelo da mesma ordem, a fim de transferir conhecimento regulatório de um fungo modelo amplamente estudado para um fungo alvo que possui poucos estudos; ii) predição de sítios de ligação de fatores de transcrição, visando encontrar os locais onde um TF se liga no DNA para ativar ou reprimir a expressão de um determinado TG (5).

Essa metodologia que utiliza ortologia e sítios de ligação de fatores de transcrição

para construção de GRNs foi utilizada para construir a GRN de 3 fungos: *Penicillium echinulatum* 2HH, *Penicillium oxalicum* 114-2 e *Fusarium oxysporum* f. sp. *lycopersici* 4287 (5) (9). Essas GRNs foram publicadas em artigos científicos na revista *Frontiers in Microbiology*, o que valida a metodologia e denota a importância da construção de GRNs para fungos (5) (9).

Figura 3 – Fontes de dados para construção de GRNs.

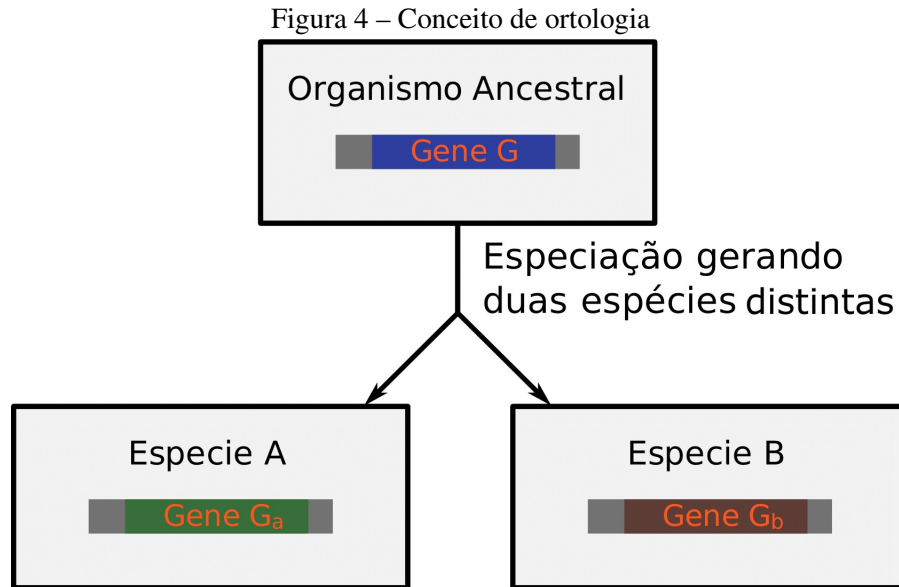


Fonte: Adaptada de (13)

### 2.1.3.1 Ortologia

Ortologia é um termo usado na biologia molecular para descrever a relação evolutiva entre genes homólogos que divergiram a partir de um ancestral comum por especiação, conforme Figura 4. De forma mais simples, ortologia refere-se à relação de homologia entre genes em diferentes organismos que possuem sua função preservada, pois compartilham um ancestral comum devido à especiação (14).

A partir do conceito de ortologia de genes, pode-se inferir que uma relação TF-TG comprovada em uma espécie poderia manter-se conservada em outra espécie filogeneticamente próxima (13). Basicamente essa relação TF-TG pode se manter, pois as duas espécies apesar de distintas na atualidade, em determinado momento histórico tiveram um ancestral em comum e conseqüentemente, podem compartilhar genes ortólogos, ou seja, com função equivalente conservada nas duas espécies (13). A ortologia de proteínas entre diferentes espécies pode ser obtida a partir de *softwares* largamente utilizados na bioinformática como *ProteinOrtho*, *OrthoMCL*, entre outros (16).

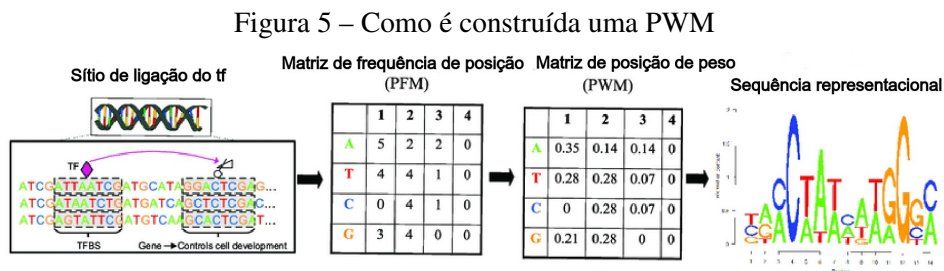


Fonte: Adaptada de (15)

### 2.1.3.2 Sítios de Ligação de Fatores de Transcrição (TFBSs)

Os TFBSs são regiões específicas no DNA onde os fatores de transcrição (proteínas que regulam a expressão gênica) se ligam. Essa ligação é fundamental para modular a atividade da transcrição, controlando assim a expressão de genes (5). Os fatores de transcrição reconhecem e se ligam a sequências de nucleotídeos específicas no DNA, geralmente chamadas de motivos ou elementos de ligação (17).

A predição de TFBSs é frequentemente realizada usando matrizes de pesos, conforme Figura 5. Essas matrizes representam modelos matemáticos das sequências de DNA que os fatores de transcrição preferencialmente reconhecem e se ligam (18, 5). Cada posição na matriz de pesos corresponde a um determinado nucleotídeo na sequência de ligação, e os valores nas posições indicam a probabilidade relativa de um determinado nucleotídeo ocorrer nessa posição.



Fonte: Traduzida de (19)

Os TFBSs podem ser preditos com o auxílio de *softwares* como RSAT, MEME, entre outros. Nota-se que existem *softwares* confiáveis e de amplo uso que são utilizados para realizar

algumas etapas necessárias para construção de uma GRN.

## 2.2 ÁREA COMPUTACIONAL

Nesta seção são abordadas algumas tecnologias e ferramentas que servem de base para o desenvolvimento do ambiente de experimentação computacional que dará suporte a nossa pesquisa. Todas as tecnologias e ferramentas apresentadas já foram utilizadas em trabalhos correlatos, o que é um indicativo potencial da viabilidade e da segurança de escolha das mesmas.

### 2.2.1 Pipeline para Construção de GRNs

Nas análises de bioinformática, um *pipeline* ou fluxo de trabalho, invariavelmente, envolve a obtenção dos dados, o gerenciamento, a organização e a transformação de diversos arquivos. Normalmente, essas transformações são feitas por softwares de terceiros, executados por linha de comando e desenvolvidos para sistemas operacionais compatíveis com Unix. O advento das técnicas de sequenciamento de nova geração, no qual milhões de sequências curtas de DNA são usadas como fonte de entrada para interpretar uma série de fenômenos biológicos, intensificou a necessidade de pipelines robustos (3). As análises de genomas tendem a envolver diversas etapas, executadas a partir da configuração de *softwares* de terceiros. Geralmente essas análises genômicas demandam alto custo computacional e muitos parâmetros de configuração (3).

O processo de construção de uma GRN demanda uma série de etapas, desde o tratamento e padronização dos dados biológicos de entrada, execução de cada uma das ferramentas de terceiros, até a própria construção da GRN. *Pipelines* podem ser implementados de diversas formas como, por exemplo, a partir de uma aplicação Web ou via linha de comando (diretamente pelo terminal no Unix). Geralmente os *pipelines* demandam arquiteturas robustas capazes de: (i) gerenciar a execução de diversas ferramentas de terceiros escritas em diferentes linguagens de programação; (ii) executar processos paralelos e sincronizados, utilizando estruturas como filas, pilhas, etc.; (iii) prover interfaces de comunicação transparentes que facilitem sua utilização por profissionais de ciências da vida; (iv) garantir a integridade dos dados contidos em arquivos intermediários de cada etapa do processo (3).

Lenz et al. (2020) propuseram uma metodologia para inferência de GRNs para fungos com base em relações de ortologia, combinadas com predição de TFBSs. Inicialmente, GRNs de

fungos modelo (Ex.: *Saccharomyces cerevisiae*) permitem o mapeamento de genes ortólogos e suas respectivas interações (TF-TG) ortólogas em um novo fungo que está sendo estudado. Em seguida, a predição de TFBSs fornece maior precisão às relações TF-TG identificadas na etapa anterior.

#### 2.2.1.1 NCBI - GenBank®

O *National Center for Biotechnology Information* (NCBI) fornece um grande conjunto de recursos on-line para informações e dados biológicos, incluindo o banco de dados de sequências de DNA (*GenBank*®) e o banco de dados PubMed® que contém artigos publicados em periódicos de ciências da vida. O sistema Entrez fornece operações de busca e recuperação para a maioria desses dados em 34 bancos de dados distintos (7).

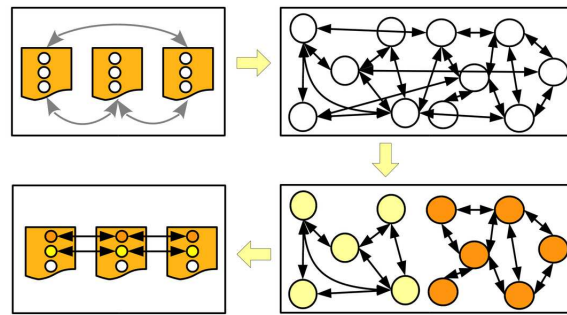
Em (18/10/2022), o GenBank® possuía um total de 12.128 genomas completos de fungos disponíveis para o uso da comunidade científica. A disponibilidade pública destes genomas é fundamental para a concepção da arquitetura proposta nesta pesquisa, levando em consideração que o usuário do sistema poderá escolher um fungo cujo genoma esteja disponível no GenBank para construir a sua GRN.

#### 2.2.1.2 ProteinOrtho

A análise de ortologia é uma importante área da bioinformática que serve de suporte para outras esferas da biologia como a genômica comparativa e as análises filogenéticas. Com o crescimento exponencial de dados de sequenciamento e conseqüentemente de genomas para serem analisados, há um crescimento na demanda de ferramentas eficientes para a identificação de proteínas ortólogas.

O ProteinOrtho é uma ferramenta autônoma, utilizada para realizar a análise de grandes conjuntos de sequências de proteínas codificadas por genes, com o objetivo de detectar genes ortólogos entre diferentes espécies (20), (21). Conforme apresentado na Figura 6, inicialmente a ferramenta realiza uma varredura entre todas as proteínas de todas as espécies (arquivos) fornecidas pelo usuário, para encontrar as proteínas similares e organizá-las em formato de grafo. Após essa fase, é iniciado o processo de agrupamento entre proteínas semelhantes, em grupos de genes ortólogos, que posteriormente são organizadas por espécie (arquivo) para gerar o resultado final da ferramenta (20).

Figura 6 – Funcionamento da ferramenta ProteinOrtho



Fonte: (21)

### 2.2.1.3 CIS-BP

O CIS-BP (*Catalog of Inferred Sequence Binding Preferences*) é uma base de dados disponibilizada na Web de maneira completamente gratuita, cujo conteúdo é um vasto catálogo de TFs. Até a data desta pesquisa (31/10/2023) a base possui dados de mais de 700 espécies de plantas, fungos e animais, totalizando mais de 390.000 TFs. Todos esses dados são coletados de mais de 70 bases distintas, como Transfac, JASPAR, HOCOMOCO, FactorBook, UniProbe e de publicações científicas. O site foi criado em conjunto por Tim Hughes da Universidade de Toronto e Matt Weirauch, atualmente Professor no Departamento de Pediatria da Universidade de Cincinnati (22).

É com o apoio do CIS-BP que temos acesso às matrizes de peso (do inglês *Position Weight Matrix* (PWM)), essas matrizes são utilizadas para representar estatisticamente os motivos (do inglês *motifs*), que são pequenas sequências de nucleotídeos localizadas na região promotora da transcrição de cada gene, esses motivos também são chamados de TFBSs.

O processo de construção de uma matriz de pesos é extenso e envolve uma série de passos. Inicialmente é necessário coletar determinadas sequências onde o TF se liga ao DNA, alinhar as sequências e realizar uma contagem de cada um dos nucleotídeos (A - Adenina, G - Guanina, C - Citosina e T - Timina) nas posições em que cada um ocorre gerando uma *Position Frequency Matrix* (PFM) (19). Com o resultado da PFM são realizados alguns cálculos matemáticos para enfim gerar as PWMs, conforme o processo mostrado na Figura 5.

### 2.2.1.4 RSAT (*Regulatory Sequence Analysis Tools*) - *matrix-scan*

Criada no ano de 1998, a ferramenta *Regulatory Sequence Analysis Tools* (RSAT) é um sistema com diversos módulos que, a depender da necessidade de cada usuário, podem

trabalhar de maneira integrada ou individualizada. A ferramenta pode ser utilizada diretamente via Web, API ou instalada em um servidor local. Seu objetivo é detectar e analisar elementos cis-regulatórios em sequências genômicas e seus principais módulos são divididos em I - Descoberta de motivos, II - Varredura de motivos, III - Análises de motivos, IV - Análises de variações regulatórias e V - Comparação de genomas (23) (24).

Em nossa pesquisa, apenas o módulo *matrix-scan* foi utilizado. Esse módulo, escrito na linguagem Perl, é responsável por analisar a região promotora de um TG avaliando-a em relação a uma matriz de peso de um TF, para encontrar possíveis TFBSs, que são regiões onde o TF se liga ao DNA da região promotora para ativar ou reprimir a expressão do TG (23).

#### 2.2.1.5 Python

*Python* é uma linguagem largamente utilizada para as mais diversas propostas de soluções de *software*, seja no ambiente comercial para o desenvolvimento de soluções que vão desde sites até análise de grande volume de dados, seja no ambiente acadêmico para realização de pesquisas das mais variadas áreas (25, 26). Por se tratar de uma linguagem de código aberto, multiplataforma, multiparadigma e com uma proposta de ser de fácil adaptabilidade e aprendizagem (27), *Python* vem conquistando cada vez mais adeptos e reconhecimento.

Criada em 1989 por Guido van Rossum a linguagem é bastante conhecida por sua baixa curva de aprendizado e sintaxe capaz de facilitar o processo de leitura do código (28). Durante os anos 2000 a linguagem foi recebendo inúmeras atualizações e bibliotecas (como por exemplo, *Biopython*, NumPy, Pandas e diversas outras) capazes de ajudar cientistas no desenvolvimento de suas pesquisas. No final dos anos 2000, se tornou a principal linguagem de programação na área de bioinformática (29, 30).

A biblioteca *Biopython* possui uma série de módulos, todos com o objetivo de facilitar algum aspecto recorrente da bioinformática. Por exemplo, leitura e escrita de diversos formatos de arquivos como FASTA, UniGene, SwissProt, alinhamento de sequências, acesso a banco de dados de biologia online e diversos outros módulos (31).

Dessa maneira o conjunto de bibliotecas e ferramentas gratuitas disponíveis, juntamente com a facilidade de leitura de código, que resulta em uma aplicação de fácil manutenibilidade, tornou a linguagem *Python* ideal e necessária para a construção da ferramenta

*FunRegulation*.

### 2.2.1.6 PostgreSQL

Para efetuar o armazenamento de todos os dados importantes para o *FunRegulation*, como usuários cadastrados, genes, organismos, proteínas, tarefas e etc, foi escolhido o banco de dados PostgreSQL. O PostgreSQL é um banco de dados de código aberto do tipo relacional que utiliza a linguagem SQL para a realização de suas operações (32). Possui mais de 35 anos de história de desenvolvimento, sendo criado em 1986 em um projeto feito na *University of California em Berkeley* nos Estados Unidos (32).

Possui uma boa reputação, sendo amplamente reconhecido como um banco de dados confiável, possuidor de um conjunto robusto de recursos e com uma dedicada comunidade de desenvolvimento que propõe soluções inovadoras e de desempenho consistente. Sua utilização pode ser feita em todos os principais sistemas operacionais da atualidade (32).

Outras soluções de banco de dados poderiam ser utilizadas sem grandes problemas, como MySQL e SQLite. Porém, alguns pontos-chave fizeram diferença na decisão final do projeto: i) PostgreSQL possui alta afinidade com o *Django* (alguns tipos de dados que o Django oferece só funcionam com o PostgreSQL); ii) seu poder de escalabilidade com altas cargas de trabalho; iii) recomendação direta do MSc. Eduardo Balbinot, consultor de tecnologias do projeto.

## 2.2.2 Gerenciamento e Escalonamento de Tarefas

A proposta de solução implementada neste trabalho de conclusão de curso, tem como objetivo ser disponibilizada via Web. Esse objetivo se dá por questões de facilidade e comodidade para os futuros usuários, para que eles não precisem se preocupar com sistema operacional, instalação de componentes e/ou configurações de *softwares*.

No entanto, ao realizar uma implementação de solução com foco na Web, outros problemas e considerações devem ser analisados com mais atenção. Uma das principais problemáticas que ocorrem na Web é a questão de lidar com múltiplos usuários e requisições de tarefas simultâneas à mesma aplicação. Destacando-se a necessidade de uma arquitetura robusta com banco de dados e outros componentes do software capazes de realizar o gerenciamento

de diversas requisições com diferentes prioridades e tempos de execução, pois uma arquitetura Cliente x Servidor comum possui problemas de evolução de arquitetura, escalabilidade e etc (33).

Para o tratamento da problemática de diversas tarefas e requisições simultâneas, é necessário criar uma arquitetura que seja capaz de gerenciar as tarefas requeridas pelos os usuários. Atualmente existem diversos *frameworks* no mercado como *Celery*, *Redis*, *Huey*, *RabbitMQ*, com o intuito de não só gerenciar requisições, mas também de facilitar o desenvolvimento do software pelo programador, conforme visto na revisão sistemática.

#### 2.2.2.1 *Celery*

O *Celery* é um *framework* de código aberto, extremamente leve, flexível, rápido e de simples implementação, capaz de lidar com o gerenciamento e escalonamento de requisições simultâneas de tarefas. Escrito na linguagem *Python* (27), o *Celery* serve como apoio, visando facilitar o desenvolvimento de tarefas que necessitam ser executadas de maneira assíncrona (34).

A fim de superar o obstáculo causado por tarefas demasiadamente lentas, basicamente o *Celery* trabalha utilizando filas de mensagens, onde uma determinada requisição de um usuário se transforma em uma mensagem, que irá para uma fila específica (35). Uma vez que a mensagem é criada, entrará em ação um módulo intermediário denominado de *broker*, cuja principal função é enviar essa mensagem para uma fila, que é constantemente monitorada por um *worker*. O *worker* é um método que tem como responsabilidade executar, de fato, o que foi solicitado pelo usuário ao sistema (34).

#### 2.2.2.2 *RabbitMQ*

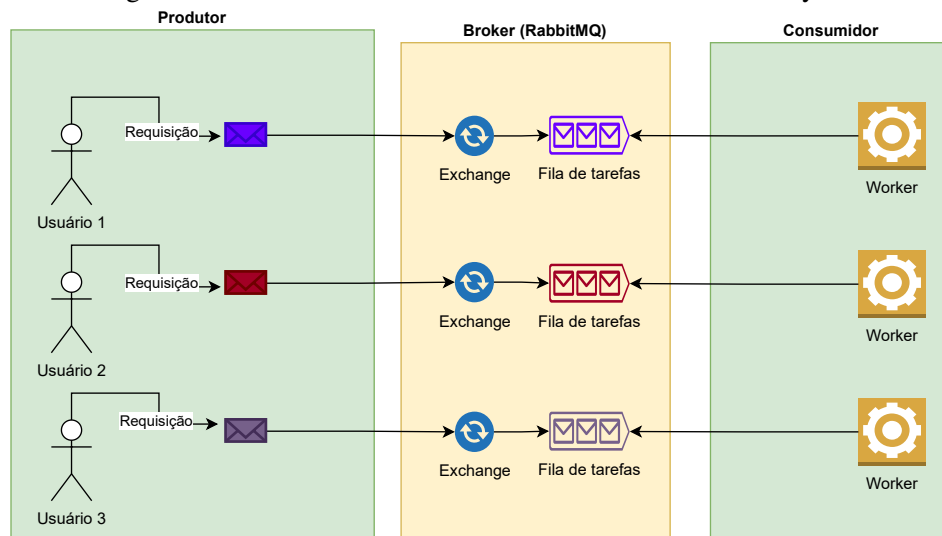
O *RabbitMQ* é um *framework* de código aberto, que foi escrito na linguagem de programação Erlang e foi lançado no ano de 2007 (36) p.4-6. O *RabbitMQ* funciona como um intermediador entre o *producer* (usuário da aplicação que produz uma requisição no sistema) e o *consumer* (método responsável pela execução da requisição do usuário) (37).

O *RabbitMQ* tem como função receber, armazenar e enviar as mensagens do *producer* para suas respectivas filas. Esse processo é realizado através das *Exchanges*, que ao receberem as mensagens irão determinar a maneira que será feito o roteamento para as filas (37).

A escolha deste *framework* ocorreu pois, além de estar presentes em outros trabalhos correlatos e ter sido recomendado por especialistas da área, a própria documentação do Celery informa que o RabbitMQ é o melhor intermediário, até o presente momento, para lidar com mensagens grandes (34).

Na figura abaixo é apresentada uma evolução da figura 7, onde os *softwares* Celery e RabbitMQ estão trabalhando em conjunto

Figura 7 – Funcionamento de um sistema de filas com Celery e RabbitMQ



Fonte: O autor, baseado em (37)

## 2.3 TRABALHOS CORRELATOS

Nesta seção são apresentados alguns trabalhos correlatos, com o objetivo de apresentarmos as similaridades e diferenças entre o *FunRegulation* e os trabalhos disponibilizados para a comunidade científica.

### 2.3.1 Seleção de trabalhos correlatos

Para a construção deste trabalho, foi necessário realizar uma revisão sistemática na literatura. O objetivo da revisão sistemática foi realizar um levantamento de tecnologias e pesquisas de ferramentas ou modelos, que tinham como foco a construção de GRNs. Desta maneira seria possível identificar possíveis lacunas de pesquisa, que pudessem servir de base para o desenvolvimento de uma proposta de Trabalho de Conclusão de Curso - TCC. A *string* de busca dos trabalhos ficou com a seguinte configuração: "*gene regulatory network*"AND ("*web*

*server"OR "web interface"OR "network visualization")*.

Ao todo foram definidos 7 critérios para a seleção de trabalhos correlatos, sendo 2 critérios de inclusão e 5 de exclusão. O primeiro critério de inclusão foram artigos recomendados diretamente pelo orientador e o segundo artigos que abordassem a construção, visualização ou inferência de GRNs. Os critérios de exclusão de artigos foram artigos duplicados, artigos sem implementação tecnológica, não relacionados com GRNs, sem visualização de GRNs e que não tivessem alguma interface.

### 2.3.2 Trabalhos selecionados

Os principais artigos e ferramentas selecionados e que inspiraram este trabalho podem ser vistos na Figura 8. No artigo (8) temos a ferramenta MALBoost, uma ferramenta WEB desenvolvida em Python e Flask em conjunto com Celery e Redis. MALBoost foi desenvolvida exclusivamente para a realização de estudos sobre o *Plasmodium falciparum*, vírus causador da doença da malária.

Outra ferramenta encontrada na revisão sistemática é a GRAND (38), um banco de dados de modelos de GRNs. GRAND foi desenvolvida utilizando Python e Django, e sua principal utilidade é para comparação de redes e predição de medicamentos. No artigo (39) é explicado sobre a Mergeomics 2.0, uma ferramenta WEB baseada em um pacote open source chamado Mergeomics R. A Mergeomics 2.0 possui diversas ferramentas inclusas implementadas como um *pipeline*, para realização de estudos de doenças e predição de medicamentos.

A principal ferramenta que serviu de base para a construção deste trabalho foi a FunRegulation (5). Criada como trabalho de doutorado de Alexandre Rafael Lenz, a *FunRegulation* adotou como tecnologias Python, Biopython e o banco de dados SQLite, e inferiu GRNs dos fungos *Penicillium ucsense*, *Penicillium oxalicum* e *Fusarium oxysporum*. Para a construção das GRNs criou a metodologia que utiliza a ortologia e o descobrimento dos sítios de ligação dos fatores de transcrição. A principal limitação do *software* se dava pelo seu funcionamento totalmente local, através de linhas de comando.

Com base nos trabalhos citados anteriormente, podemos observar uma série de lacunas que podem ser mitigadas. Por exemplo, algumas ferramentas servem apenas para um organismo específico, enquanto outras servem para a compreensão de doenças e predição de

medicamentos ou não tem foco para estudo de fungos. Desta maneira, este trabalho pode atuar evoluindo além da área de estudos de GRNs, mas também a área de estudo de fungos. Visto que poucas ferramentas de criação de GRNs atuam visando a compreensão dos fungos.

Figura 8 – Lista com os trabalhos correlatos

Ferramenta	Alvo(s)	Tipo de aplicação	Tecnologias	DOI	Comentários
<i>FunRegulation</i>	<i>Penicillium ucsense</i> , <i>Penicillium oxalicum</i> e <i>Fusarium oxysporum</i>	Linha de comando UNIX	Python, BioPython e SQLite	10.3389/fmicb.2020.588263 e 10.3389/fmicb.2022.861528	O processo de construção da GRN não é totalmente automatizado.
<i>MALBoost</i>	<i>Plasmodium falciparum</i>	WEB	Python, Flask, Redis e Celery	10.1186/s12936-021-03848-2	Só permite a construção da GRN para o Plasmodium.
<i>GRAND</i>	Doenças	WEB	Python, Django	10.1093/nar/gkab778	Construção de GRNs para compreensão de doenças.
<i>Mergeomics 2.0</i>	Doenças	WEB	R	10.1093/nar/gkab405	Construção de GRN para entedimento de doenças.

Fonte: O autor

### 3 METODOLOGIA

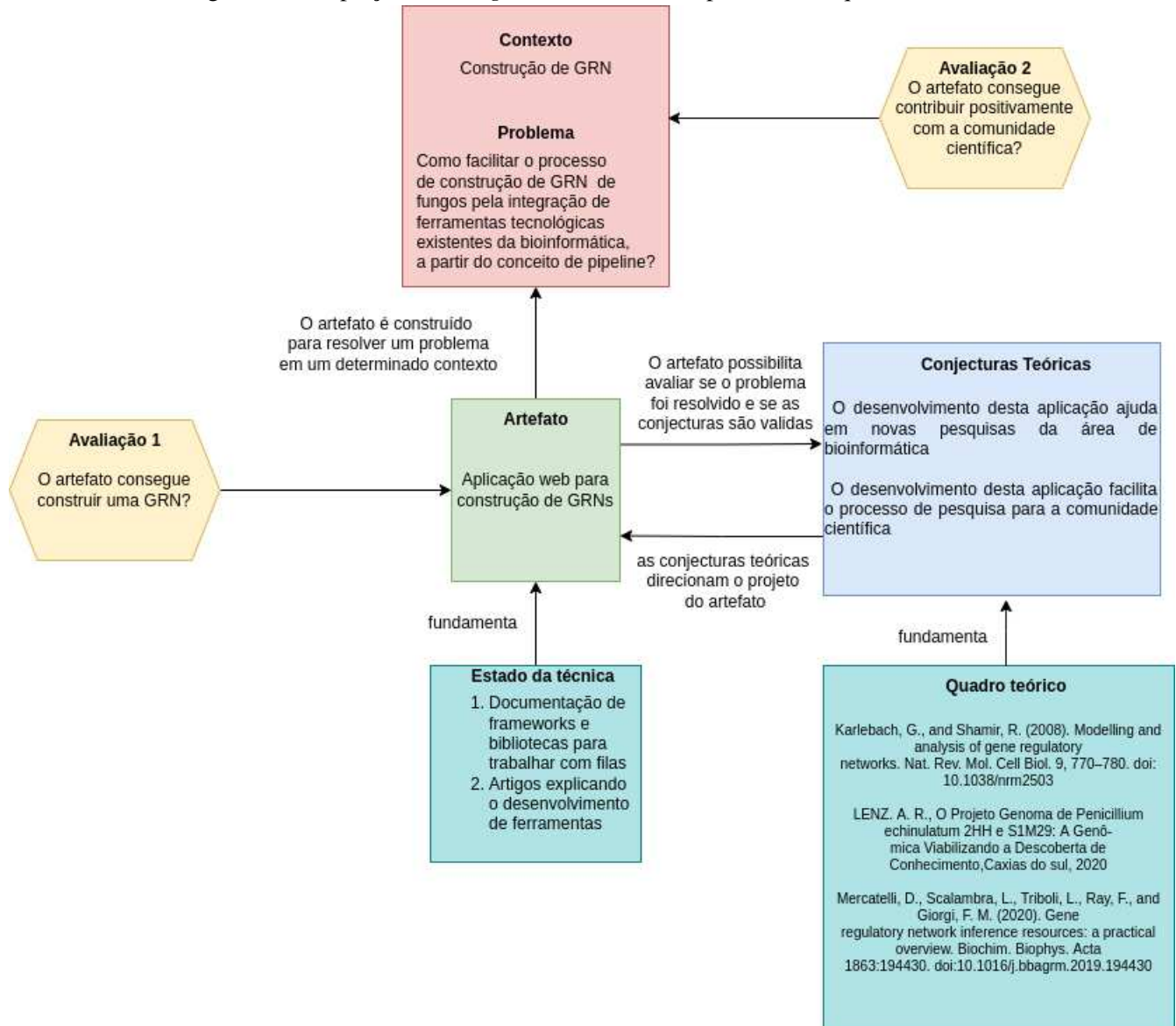
Neste capítulo apresentamos as metodologias de pesquisa e de desenvolvimento deste trabalho. A metodologia de pesquisa serve como alicerce científico do trabalho, ao passo que, a metodologia de desenvolvimento engloba o aspecto prático do trabalho. Quanto a natureza deste trabalho possui a categoria de pesquisa aplicada.

A pesquisa aplicada objetiva gerar conhecimentos para aplicação prática, dirigidos à solução de problemas específicos, que envolvem verdades e interesses locais. No caso desta pesquisa em questão, observamos uma lacuna, citada na seção introdutória, que reflete a falta de ferramentas para a construção de GRNs de fungos. Para contribuir com esse problema, vamos identificar as ferramentas existentes e definir uma arquitetura. A arquitetura deve ser capaz de preencher essa lacuna e conseqüentemente, ser utilizada para futuras pesquisas no campo da bioinformática, mais especificamente, estudos que envolvam fungos.

Quanto aos objetivos essa pesquisa se classifica como exploratória, pois tem como finalidade proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito ou a construir hipóteses. Essa pesquisa se encaixa como pesquisa exploratória pois, foi necessário realizar um levantamento de artigos que abordam ferramentas para construção de GRNs, e consulta com especialistas na área, para que fosse possível utilizar o conhecimento já existente e adaptar para a concepção da arquitetura para construção de GRNs para fungos.

Quanto aos procedimentos este trabalho utiliza a *Design Science Research* (DSR). A DSR é uma abordagem de pesquisa que se concentra na criação de artefatos para resolver problemas práticos. Essa abordagem é frequentemente aplicada em disciplinas como ciência da computação, sistemas de informação, engenharia e gestão. A escolha do método foi bastante natural pois o principal objetivo da DSR é desenvolver soluções práticas e úteis para problemas do mundo real: 1) desenvolver um artefato para resolver um problema prático em algum contexto; 2) gerar novos conhecimentos técnicos e científicos. É exemplificado na figura 9, aspectos importantes do método DSR. Ao centro da figura 9 se encontra o artefato que construímos nesta pesquisa e ao redor do artefato atributos importantes, como avaliações, conjecturas teóricas, a problemática que motiva o trabalho e etc.

Figura 9 – Adaptação da *Design Science Research* para esta Pesquisa



Fonte: O autor, baseado em (40)

O trabalho definiu duas métricas avaliativas para o artefato construído nesta pesquisa. Primeiramente foi definida a avaliação 1 exposta na figura 9 que consiste da seguinte questão “O artefato consegue construir uma GRN?”, sendo definido um objetivo específico para responder esta questão. O objetivo compreende gerar duas GRN de fungos de ordens distintas, para gerar duas rotas de *pipeline* (com e sem RSAT) e dessa maneira, obtemos os dados necessários para serem enviados ao *front-end*. Os dados consistem nas interações regulatórias dos fungos que serão salvos no banco de dados e posteriormente, serão dispostos em um arquivo no formato JSON que será utilizado no *front-end* para a construção do grafo da GRN e enfim realizarmos a validação.

A avaliação 2 também definida na figura 9 tem como objetivo responder a seguinte questão: “O artefato consegue contribuir positivamente para a comunidade científica?”. Para conseguirmos responder essa questão, será necessário realizarmos primeiramente a integração deste trabalho com o *front-end* e após essa integração, disponibilizarmos a *FunRegulation* para que a comunidade científica, interessada em GRNs, possa utilizar e testar a nossa aplicação. Após a utilização da nossa aplicação pelo usuário, realizaremos uma entrevista/questionário para sabermos se a aplicação de fato, consegue colaborar para futuras pesquisas na comunidade científica e também entendermos, possíveis melhorias e problemas que a nossa aplicação possa ter.

### 3.1 METODOLOGIA DE DESENVOLVIMENTO

Esta seção tem como objetivo apresentar de maneira breve e objetiva, a metodologia e algumas ferramentas adotadas para desenvolvimento do artefato (arquitetura Web (*Back-end* para construir GRNs de fungos).

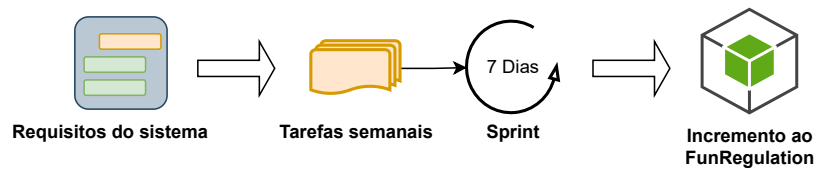
Anteriormente a este trabalho de conclusão de curso, foram definidos os requisitos funcionais (RFs) e requisitos não funcionais (RNFs) da ferramenta Web FunRegulation (apresentados de forma resumida nas seções subsequentes e de forma completa no Anexo A), em seguida foi realizada a análise dos requisitos levantados gerando um conjunto de 16 casos de uso (apresentados de forma resumida nas seções subsequentes e de forma completa no Anexo B). Por fim, foi definida a ordem de prioridade para desenvolvimento destes casos de uso, dando início a este trabalho de pesquisa.

Para desenvolvimento do artefato foi proposta uma adaptação do *framework* SCRUM (41), apresentada na Figura 10. Com base nestas prioridades dos casos de uso, as *Sprints* foram divididas em blocos de uma semana, onde ao final de cada semana foram apresentadas ao orientador as funcionalidades implementadas, a fim de validar o resultado da implementação.

O SCRUM é uma metodologia ágil, sendo mais utilizada na área de desenvolvimento de *softwares*. Um de seus principais benefícios é a sua capacidade adaptativa e sua flexibilidade, podendo ajudar times em diversos tipos de projetos complexos (42, 41).

Para realizar o acompanhamento das atividades de desenvolvimento, juntamente com o SCRUM utilizamos o quadro Kanban. O quadro Kanban é bastante utilizado para organizar

Figura 10 – Ciclo de desenvolvimento do *FunRegulation*



Fonte: O autor, baseado em (43)

as tarefas de desenvolvimento. Com este quadro é possível que o desenvolvedor ou equipe, acompanhe o progresso das tarefas que foram feitas, que estão em andamento e que estão por fazer (44). Para a gestão do quadro *Kanban* utilizamos a ferramenta on-line e gratuita chamada Trello (45), onde cadastramos e gerenciamos todas as atividades referentes ao desenvolvimento da arquitetura *Back-end* do *FunRegulation*.

Por importantes questões como segurança, *backup*, controle de versão e disponibilidade de código, foi necessário o uso de uma ferramenta de gerência de configuração. O GitHub (46) foi escolhido pelo fato de ser uma das ferramentas mais populares e confiáveis da atualidade (47). O código-fonte do *FunRegulation* está disponível publicamente para todos os interessados em entender melhor a ferramenta e que desejem utilizar alguma parte do código, através do endereço (48).

## 4 FUNREGULATION

Neste capítulo são apresentados aspectos referentes à definição do FunRegulation, aos requisitos do FunRegulation, à construção e ao funcionamento das ferramentas e *softwares* que compõem a arquitetura do *FunRegulation*.

### 4.1 O SISTEMA FUNREGULATION

O FunRegulation é uma ferramenta *desktop*, que foi desenvolvida para construção de GRNs. Atualmente o FunRegulation está em fase de evolução para Web, sendo dividido em dois trabalhos de conclusão de curso, compreendendo a construção do *Back-end* e do *Front-end*.

A parte que compete a este trabalho é uma arquitetura Web (*Back-end*) disponibilizada em forma de uma API, responsável pela geração de GRNs de fungos. Estas GRNs são geradas através de um *pipeline*, organizado e composto por diversas ferramentas de terceiros. Inicialmente o usuário deve realizar um requisição, selecionando o fungo que deseja estudar e o parâmetro, de inclusão ou não do RSAT, para a identificação dos sítios de ligação dos fatores de transcrição. Após a requisição ser realizada, o processo de construção da GRN é iniciado.

A construção das GRNs no FunRegulation é baseada em ortologia (transferência de conhecimento) e sítios de ligação de fatores de transcrição (TFBS). Com base em artigos publicados como (5) (9), para transferir conhecimento regulatório de um organismo para outro, inicialmente é necessário que se tenha uma base de dados de informações regulatórias dos organismos modelo:

- Ordem *Eurotiales* - *Aspergillus nidulans* FGSC A4
- Ordem *Hypocreales* - *Neurospora crassa* OR74A
- Ordem *Saccharomycetales* - *Saccharomyces cerevisiae* S288c
- Ordem *Sordariales* - *Fusarium graminearum* PH-1

Organismos modelo são espécies de organismos que são estudadas de forma intensiva em pesquisa científica, geralmente porque possuem características específicas que tornam mais fácil investigar fenômenos biológicos, processos celulares, genéticos ou fisiológicos. Esses

organismos são escolhidos por sua relevância para responder a questões específicas ou representar fenômenos mais amplos de interesse científico. *Saccharomyces cerevisiae*, por exemplo, é uma levedura muitas vezes utilizada na produção de alimentos e bebidas, como pães, iogurtes e cerveja.

Em seguida, foram selecionados todos os genomas completos publicados no *GenBank* de fungos que pertencem às 4 ordens citadas, totalizando 1.361 fungos. As informações genômicas (números de acesso, ordem, gênero, espécie e linhagem) foram carregadas no banco de dados do sistema.

Atualmente existem cadastrados no banco de dados 1.365 (mil trezentos e sessenta e cinco) organismos, que são fungos cujos genomas foram disponibilizados no *GenBank*, desse total apenas 4 (quatro) são os chamados organismos modelo, enquanto que o número restante são fungos alvo, que o usuário tem a opção de selecionar para construir a sua GRN.

Esses fungos foram inseridos no banco de dados, através de um *script* escrito na linguagem Python (27). O *script* realiza a leitura de um arquivo TSV que possui todas as informações para identificação de um determinado fungo. Essas informações são: código identificador do organismo (*accession*), ordem, gênero, espécie, cepa, se o organismo é modelo e se possui matrizes de peso.

No sistema desenvolvido neste trabalho, o *FunRegulation*, após o usuário selecionar dentre os 1.361 fungos disponíveis para a construção de sua GRN, o sistema irá identificar qual o seu organismo modelo correspondente, de acordo com a sua ordem.

Outra etapa importante de coleta de dados correspondeu a busca de matrizes de peso no banco de dados Cis-BP para cada um dos 1.361 fungos selecionados no GenBank. Foram encontradas matrizes de peso para os TFs de 83 fungos. Essas informações são fundamentais para a execução da etapa de predição de TFBSs, pois essas matrizes de peso, são os dados de entrada do software RSAT. Desta maneira, apenas com estes 83 organismos, que o usuário poderá habilitar a opção de predição de TFBSs.

Nas seções subsequentes são explicados com mais detalhes todo o processo de criação das GRNs.

#### 4.1.1 Levantamento de Requisitos

O levantamento de requisitos, definidos durante a fase de especificação do sistema, visa compreender quais as necessidades e implementações são necessárias, para que se atinja ao final do desenvolvimento um produto mais completo possível, capaz de atender as necessidades dos usuários de maneira eficiente e eficaz.

Conforme o documento detalhado apresentado no Anexo A que define, dentre outros elementos, os requisitos funcionais do sistema, temos como funcionalidades que irão compor o *Back-end* da solução os seguintes RFs:

- RF01 - Cadastrar usuário
- RF02 - Confirmar cadastro
- RF03 - Efetuar Log in
- RF04 - Efetuar Log out
- RF05 - Recuperar senha
- RF06 - Construir GRN
- RF07 - Carregar informações do NCBI
- RF10 - Exibir anotação funcional
- RF11 - Exibir estatísticas

#### 4.1.2 Análise de requisitos

No Anexo B encontra-se o documento detalhado com o diagrama e as descrições dos casos de uso do sistema. A seguir são apresentados os casos de uso relacionados ao *Back-end* da solução com suas respectivas prioridades:

- P5 UC01 - Manter usuário
- P6 UC02 - Confirmar cadastro
- P7 UC03 - Efetuar Log in
- P8 UC04 - Efetuar Log out
- P9 UC05 - Redefinir senha
- P2 UC06 - Construir GRN
- P1 UC07 - Obter organismo do NCBI
- P3 UC10 - Exibir anotação funcional

- P4 UC11 - Exibir estatísticas

## 4.2 IDENTIFICAÇÃO DAS ETAPAS E FERRAMENTAS PARA COMPOSIÇÃO DO PIPELINE

Conforme prioridade de desenvolvimento definida para os casos de uso na seção anterior, inicialmente foi necessário obter uma série de informações de cada organismo escolhido pelo usuário (UC07). Informações como sequências de proteínas, localização dos genes e regiões promotoras, de maneira rápida, eficaz e confiável. Sem esses dados, não seria possível realizar operações básicas para transferência de conhecimento regulatório de um organismo modelo para um organismo alvo.

O banco de dados inicialmente definido para o projeto foi o ENSEMBL, que tinha todos os dados necessários para a implementação do caso de uso 07 - “Obter organismo do ENSEMBL”. Contudo, devido a limitações tais como: falta de rotas para buscar o conjunto completo de sequências de proteínas pelo seu número de acesso, foi substituído pelo banco de dados NCBI (GenBank). O GenBank é um banco de dados público que oferece uma solução de integração simples e direta, com linguagens como Python e Javascript.

Através do GenBank é possível obtermos informações necessárias ao *FunRegulation* do fungo selecionado pelo usuário. Informações de proteínas, promotores e genes estão dispostas em dois arquivos apelidados de *genomic.gbff* e *protein.faa*. A partir do *genomic.gbff* poderemos extrair informações dos promotores, genes e das proteínas, que serão salvas no banco de dados. O arquivo *protein.faa* que contém somente informações referente às proteínas do fungo, será utilizado pelo *software* ProteinOrtho para encontrar as proteínas ortólogas entre o fungo selecionado pelo usuário e o fungo modelo.

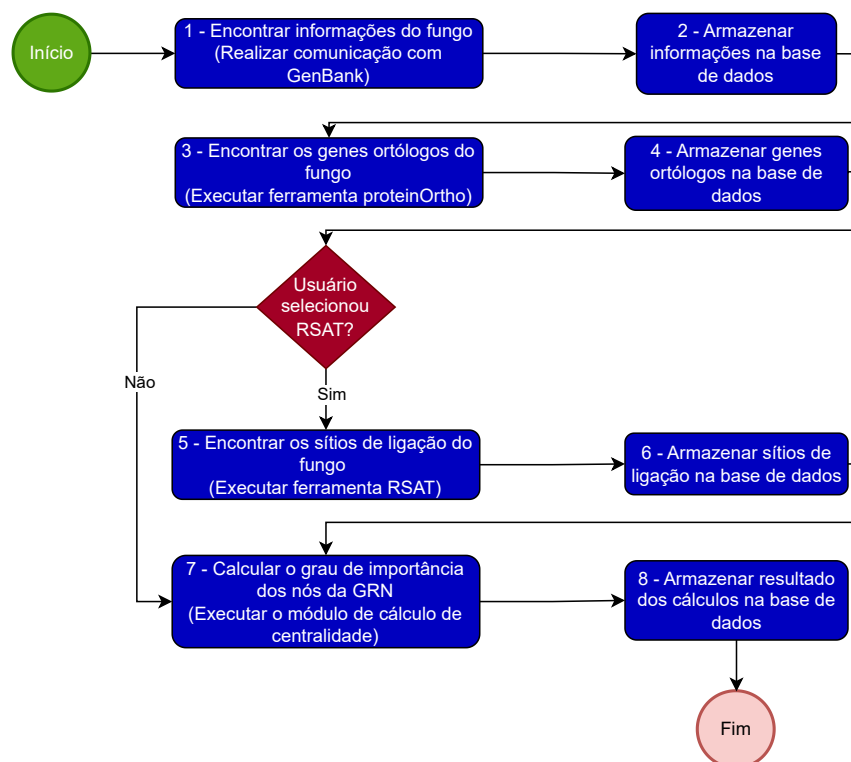
Após uma análise detalhada do caso de uso 06 - “Construir GRN”, presente no Anexo B deste trabalho, foi idealizado uma sequência de etapas necessárias (Figura 11), as quais o *software* deveria percorrer, para que fosse possível atingir o resultado esperado de construção de uma GRN.

Para a construção da GRN este trabalho utiliza-se dos conceitos de ortologia e de sítios de ligação de fatores de transcrição. A pesquisa de genes ortólogos entre dois fungos, é feita através do *software* ProteinOrtho. O *software* recebe como entrada os conjuntos completos de

sequências de proteínas do fungo modelo e do fungo alvo no formato FASTA (formato baseado em texto para representar sequências de nucleotídeos e aminoácidos), gerando como resultado a lista de ortólogos no formato *Tab-separated values* (TSV) que é carregado para o banco de dados. Já os TFBSs são encontrados através do módulo *matrix-scan* do RSAT. O *matrix-scan* recebe como entrada a região promotora de um TG e uma matriz de peso correspondente de um TF, efetuando a predição de possíveis TFBSs, os quais também são salvos no banco de dados. Ambos os *softwares* foram escolhidos pois já haviam sido utilizados e aprovados nos trabalhos que antecederam esta pesquisa.

Por fim foi adicionado um módulo de cálculo de centralidade da GRN, responsável por calcular de 5 maneiras distintas (*degree, closeness, betweenness, harmonic, eigenvector*) o grau de importância de um determinado nó da GRN.

Figura 11 – Fluxograma com os passos da construção de uma GRN



Fonte: O autor

#### 4.2.1 Instalação das ferramentas

Para realizar a instalação do *software* ProteinOrtho, primeiramente foi necessário o *download* de seus arquivos através do (21) (utilizando o comando WGET), onde está dispo-

nibilizado um guia para a realização de todo o processo de instalação e sua utilização. Para executarmos a ferramenta é necessário a linguagem de programação Perl juntamente com seus módulos e o programa BLAST.

Para a instalação da ferramenta RSAT é necessário que o sistema operacional seja uma distribuição linux, principalmente o Ubuntu, pois em outros sistemas também baseados no linux são necessárias alterações no processo de instalação ou sistema operacional MacOS. Todo o código do RSAT está disponibilizado no Github (24) e todo o processo de instalação está documentado no (49), sendo necessário clonar 4 repositórios para o funcionamento completo da ferramenta, sendo os repositórios *rsat\_code* (código da ferramenta), *demo\_files* (arquivos para testes), *motif\_databases* (banco de dados com os motifs de ligação de TFs) e *sample\_outputs* (conjuntos de amostras com resultados da ferramenta).

Apesar da documentação estar bem organizada, mostrando um passo a passo sobre o que deve ser executado, a documentação é falha na questão de erros que podem ocorrer durante o processo de instalação. Por exemplo, existe um trecho de instalação onde é necessário executar 9 arquivos de configuração e instalação de ferramentas que servem de suporte ao RSAT (Python, R, Perl, configuração de servidor APACHE e etc), onde caso ocorram erros, não é exibido o erro e/ou motivo do erro sendo necessário analisar arquivos de log de instalação da ferramenta.

#### **4.2.2 O funcionamentos das etapas e ferramentas que compõem o *pipeline***

Após o usuário enviar a solicitação de construção de uma GRN para o FunRegulation, é iniciado o fluxograma da Figura 11. No passo 1 é realizado o *download* das informações genômicas do organismo solicitado pelo usuário na requisição no GenBank. O *download* destas informações genômicas são feitas através de uma comunicação via API, sendo que o código responsável por essa comunicação, é disponibilizado pela própria equipe de tecnologia do NCBI que mantém o GenBank. Foram necessárias poucas alterações no código, apenas para adaptar mensagens de log e o caminho para armazenamento de arquivos.

São baixados dois tipos de arquivos, sendo o primeiro com uma extensão .GBFF (*GenBank Flat File*), onde são encontradas uma diversidade de informações como proteínas, genes, sequências nucleotídicas e outros dados genômicos. Tais informações são utilizadas no passo 2 do fluxograma, sendo salvas no banco de dados nas tabelas *gene*, *promoter* e *protein*. O segundo arquivo possui uma extensão .faa, que é um arquivo FASTA contendo as sequências

de proteínas do organismo. Este segundo arquivo é utilizado no passo 3 do fluxograma, para a execução do programa ProteinOrtho, responsável por encontrar os genes ortólogos entre o fungo modelo e o fungo alvo. Ao final da execução do ProteinOrtho é gerado um arquivo .tsv contendo os genes ortólogos, sendo este arquivo lido e suas informações armazenadas no banco de dados conforme o passo 4 do fluxograma.

Ao final do passo 4 do fluxograma de construção da GRN, o *FunRegulation* entra em um ponto de decisão: a inclusão ou não do *software* RSAT. O RSAT pode não ser incluído no *pipeline* por dois motivos, o primeiro se refere ao próprio organismo selecionado pelo usuário, pois este organismo pode não possuir matrizes de peso para alimentar a ferramenta. O segundo motivo é que mesmo em casos onde o organismo possui os dados necessários, o usuário pode escolher não executar essa ferramenta. Caso o RSAT seja incluído no *pipeline*, então o *software* é executado, conforme o passo 5 do fluxograma, para descobrir possíveis TFBSs, que são porções do gene onde o um TF pode se ligar para ativar ou reprimir a expressão de um gene. Ao final da execução do RSAT, as informações encontradas são inseridas no banco de dados, conforme o passo 6 do fluxograma.

Caso o RSAT não seja incluído no pipeline, os passos 5 e 6 são ignorados, sendo então executados os passos de número 7 e 8. O passo 7 é referente ao cálculo de centralidade dos nós da GRN. Esse é um módulo de autoria de Evelyn do grupo de pesquisa G2BC (Grupo de Pesquisa em Bioinformática e Biologia Computacional) da UNEB, que possui como objetivo informar o grau de importância de cada nó presente em uma GRN. Para isso são realizados 5 cálculos chamados de *degree*, *closeness*, *betweenness*, *eigenvector* e *harmonic*. No cálculo *degree* basicamente é realizada uma contagem de quantos relacionamentos um determinado nó possui na GRN, enquanto que os cálculos restantes são modelos matemáticos mais complexos. O resultado de cada um desses cálculos varia entre 0 e 1 (maior importância possível).

O módulo de cálculo de centralidade recebe como entrada todos os TFs e os TGs da GRN do fungo selecionado pelo usuário. A partir dos TFs e TGs é criado um grafo através da biblioteca *Networkx* com o método *DiGraph*, que será utilizado como entrada para as funções de cálculo de centralidade disponibilizados também através da biblioteca *Networkx* e de seus módulos *algorithms* e *centrality*. As funções são *degree\_centrality*, *closeness\_centrality*, *betweenness\_centrality*, *eigenvector\_centrality*, *harmonic\_centrality*. O passo 8 e último passo do *pipeline* insere todos os resultados dos cálculos de centralidade no banco de dados, na tabela

*calculatecentralityregistry*.

### 4.3 DESENHO DA ARQUITETURA

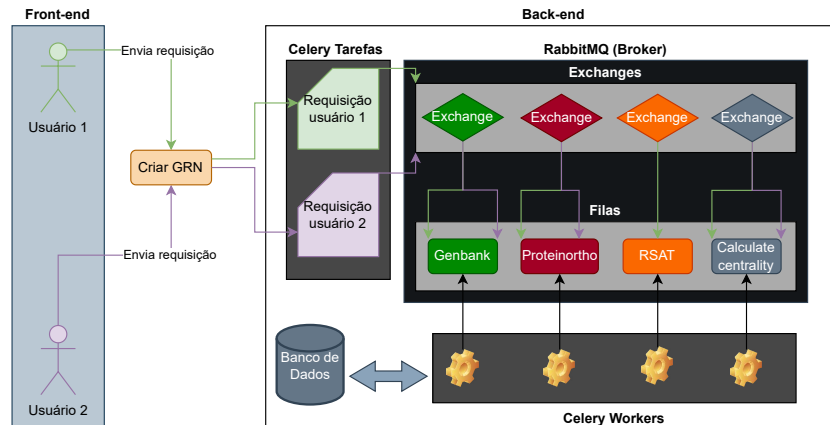
A definição da solução lida com diversos desafios, tais como, a responsabilidade de gerenciamento de múltiplas tarefas, várias dessas partindo de diferentes usuários e/ou diferentes ferramentas de terceiros, confiabilidade na obtenção de dados dos fungos, instalação local de ferramentas de terceiros. Esses detalhes mostram o nível de robustez necessário da nossa arquitetura para possibilitar o processamento de todas essas demandas.

A solução projetada utilizou como referência outras ferramentas que também fazem a construção de GRNs para que fosse possível entender alguns aspectos importantes de construção dessas ferramentas, como linguagens de programação utilizadas, estrutura de dados, *frameworks* utilizados, etc. Todo o aparato tecnológico foi escolhido, baseado em trabalhos anteriores, pesquisas nas documentações e comunidades de cada uma dessas tecnologias. A arquitetura (*Back-end*) proposta é demonstrada na Figura 12, com duas rotas de execução possíveis para o *pipeline*, sendo que o *pipeline* definido possibilita a execução de duas rotas, utilizando ou não o RSAT. na Figura 12 temos o usuário 1 representado pela cor verde, solicitando a inclusão do RSAT no *pipeline*, enquanto que o usuário 2 representado pela cor roxa, não inclui o RSAT no *pipeline*.

A solução compreende os seguintes componentes tecnológicos:

- Python
- Django (com a biblioteca Django REST *Framework*)
- Celery
- RabbitMQ
- PostgreSQL
- Estrutura de filas para gerenciamento das *tasks* dos usuários

Utilizamos Python por se tratar de uma linguagem com diversas bibliotecas disponíveis e por diversos trabalhos vistos na revisão sistemática, também optamos por utilizar dessa linguagem de programação muito poderosa e de fácil leitura de código. O Django foi utilizado para facilitar a criação de uma API REST, é uma interface que dois sistemas de computador usam para trocar informações de forma segura pela internet. O Django oferece uma variedade de

Figura 12 – Arquitetura (*Back-end*) do Sistema FunRegulation

Fonte: O autor

bibliotecas que tem o poder de facilitar o desenvolvimento, como no caso da biblioteca Django REST *Framework* que em poucas linhas de código, é capaz de gerar *endpoints* e métodos de CRUD (*Create, Read, Update e Delete*).

Algumas tarefas existentes no *pipeline* da construção de uma GRN podem levar diversos minutos para serem executadas, até finalmente terem um resultado para ser exibido para o usuário. Esse problema de escalabilidade pode piorar, se levamos em consideração que o sistema pode ter múltiplos usuários gerando diversas requisições simultâneas ao sistema. Então foi necessário desenvolver uma solução viável, capaz de tratar essa problemática e, conseqüentemente, oferecer uma melhor experiência ao usuário.

Para trabalhar com as listas de tarefas, o *framework* Celery foi escolhido para prover uma arquitetura de *tasks* (tarefas do cliente) e *workers* (serviços que ficam monitorando a fila para executar uma determinada *task*). Essa comunicação entre *tasks* e *workers* é feita através do RabbitMQ, *software* responsável por realizar o roteamento de tarefas para suas respectivas filas. A escolha desses *frameworks* foram embasadas por trabalhos correlatos, pelo tamanho da comunidade que cada uma possuía e também com realização de consultas com pesquisadores e especialistas na área.

Cada tarefa do nosso *software* (*download* de arquivos do GenBank via *web service*, chamada de programas de terceiros, etc.) é denominada *task*. Para realizar o controle das diversas *tasks* do programa e evitar que o usuário tenha problemas com *tasks* muito longas e possíveis erros de *timeout*, é utilizado o *framework* Celery. O Celery é responsável por criar uma lista de *tasks* e consumi-lás com os chamados *Workers*, que verificam constantemente as filas de *tasks*.

Para que possa ocorrer a comunicação entre as *tasks* solicitadas pelos usuários e os *workers*, é necessário um mecanismo denominado *Message Broker*. O *Broker* é o responsável por inserir as *tasks* em suas respectivas filas, sendo o RabbitMQ, o *broker* escolhido para este projeto.

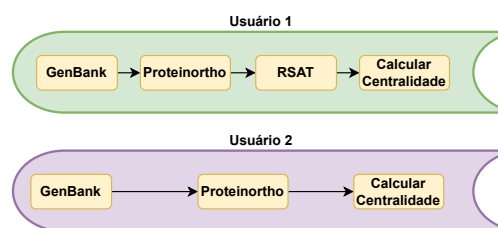
### 4.3.1 Implementação do Gerenciamento e Escalonamento de Tarefas

Nesta seção é explicado o funcionamento das etapas da arquitetura do *FunRegulation*, detalhes da implementação e os *end-points* criados para comunicação com o *front-end*.

#### 4.3.1.1 Lógica de Encadeamento das *tasks*

As tarefas no *pipeline* da *FunRegulation*, precisam seguir uma ordem específica para a construção da GRN. A necessidade de criar uma sequência lógica de tarefas, ocorre pelo fato de que o resultado gerado em uma determinada tarefa, são os dados que uma outra tarefa precisa ter acesso para executar corretamente suas funções. Por exemplo, não é possível executar a ferramenta ProteinOrtho, para encontrar as proteínas ortólogas do fungo sem antes ter disponível o arquivo de proteínas do fungo que foi previamente selecionado pelo usuário do sistema. Caso não seja feito esse encadeamento, a ordem de execução das tarefas ocasionará erros na execução do *pipeline*.

Figura 13 – Rotas de Execução do Pipeline de dois usuários distintos



Fonte: O autor

Para vencer esse obstáculo foi utilizado um método disponibilizado pelo Celery chamado de *chain* (corrente, em tradução direta). O método *chain* tem como parâmetro as assinaturas das tarefas que serão executadas sequencialmente e cada uma das tarefas, presentes na “corrente de tasks” será chamada via uma função *callback*. Uma *callback* é resumidamente uma função que só será chamada quando uma outra tarefa terminar completamente sua execução. Na Figura 13 podemos ver o exemplo de duas possíveis rotas do *pipeline* do *FunRegulation*,

onde uma tarefa executa via *callback* uma outra tarefa, seguindo assim uma sequência lógica de tarefas.

#### 4.4 IMPLEMENTAÇÃO DOS REQUISITOS - *END-POINTS*

Para a construção da arquitetura do *FunRegulation* foram desenvolvidas uma série de endpoints que compõem uma API REST construída com o Django REST *Framework*. Cada *end-point* tem o objetivo de atender a uma determinada funcionalidade e/ou demanda do usuário durante a utilização do sistema. A descrição e o objetivo dos *end-points* são destacados nas subseções seguintes.

##### 4.4.1 Listar organismos - RF07

Este *end-point* tem como objetivo listar todos os organismos (com exceção dos organismos que são modelos), que estão presentes em nossa base de dados. Este *end-point* será utilizado para popular uma *dropdown* com todos os fungos alvo que o usuário pode escolher para construção de sua GRN.

Figura 14 – Retorno do *end-point* para listar organismos

```
[
  {
    "accession": "GCA_000001985.1",
    "order": "Eurotiales",
    "genus": "Talaromyces",
    "species": "marneffeii",
    "strain": "ATCC 18224",
    "is_model": false,
    "cis_bp": true
  },
  {
    "accession": "GCA_000002515.1",
    "order": "Saccharomycetales",
    "genus": "Kluyveromyces",
    "species": "lactis",
    "strain": "NRRL Y-1140",
    "is_model": false,
    "cis_bp": true
  }
]
```

Fonte: O autor

Na Figura 14 temos um exemplo do retorno no formato JSON (*JavaScript Object Notation*) de dois organismos. Os dados retornados são código de acesso do organismo (*acces-*

sion), ordem, gênero, espécie, cepa (*strain*), um booleano indicando se o organismo é modelo ou não no campo *is model* e um outro campo *cis bp* também booleano indicando se o organismo possui matrizes de peso ou não.

#### 4.4.2 Criar uma GRN - RF06

Este é o *end-point* principal do *FunRegulation*, pois é a partir deste, que todo o processo do pipeline é iniciado. No momento em que o usuário enviar ao sistema, a requisição para construir a GRN do fungo selecionado, será executada a sequência de tarefas, conforme os parâmetros selecionados pelo usuário. Na Figura 15 podemos ver uma sequência de tarefas para o fungo de código de acesso GCA\_000003125.1, onde a execução do software RSAT não está incluída.

Figura 15 – Exemplo de requisição para criação de GRN

```
{
  "organism_accession": "GCA_000003125.1",
  "rsat_analyse": false,
  "proteinOrtho_analyse": true
}
```

Fonte: O autor

Após a solicitação é retornado ao usuário a confirmação das informações selecionadas anteriormente e um registro, informando a identificação da tarefa que será utilizada posteriormente para acompanhar o andamento de cada um dos passos do *pipeline* de maneira mais detalhada, conforme a Figura 16.

Figura 16 – Exemplo de retorno da requisição para criação da GRN

```
{
  "registry": 347,
  "request": {
    "organism_accession": "GCA_000003125.1",
    "proteinOrtho_analyse": true,
    "rsat_analyse": false,
  }
}
```

Fonte: O autor

#### 4.4.3 Acompanhar as tarefas do pipeline - RF06

Conforme já citado neste trabalho, as tarefas para a construção de uma GRN podem levar um tempo razoável até a sua finalização. No ambiente local de desenvolvimento do FunRegulation (I5 5 geração, 16 ram ddr3 e HD) um *pipeline* completo (ProteinOrtho, RSAT e cálculo de centralidade) levou um pouco mais de 1 hora até a construção completa da GRN.

Por esse motivo é extremamente importante que o usuário possa acompanhar, a situação atual do *pipeline* como, o que será executado, o que está em execução ou se ocorreu algum erro. A Figura 17 exemplifica os estados que a tarefa pode possuir, o significado de cada estado e a informação que é enviada ao usuário para cada um dos estados.

Figura 17 – Exemplo de retorno de consulta de estado da tarefa

```
{
  "Order analyse": "COMPLETED",
  "Download organism": "EXECUTING",
  "ProteinOrtho": "WAITING FOR EXECUTION",
  "Creating graph": "ERROR",
  "Degree calculation": "COMPLETED",
  "Closeness calculation": "COMPLETED",
  "Betweenness calculation": "COMPLETED",
  "Eigenvector calculation": "COMPLETED",
  "Harmonic calculation": "COMPLETED",
  "RSAT": "COMPLETED",
  "Pipeline Completed": true
}
```

Fonte: O autor

#### 4.4.4 Obter informações de um nó da rede - RF10

Alguns usuários podem ter a necessidade de obter mais conhecimento sobre uma proteína específica presente na GRN. Por isso foi criado um *end-point* que tem como objetivo enviar todas as informações contidas no banco de dados sobre aquela proteína. Ao clicar sobre um nó na rede o usuário terá acesso à informações como INTERPRO, PFAM, Gene3D, etc.

#### 4.4.5 Obter medidas de centralidade de um nó da GRN - RF11

As informações sobre o grau de centralidade de um determinado nó da GRN podem ser importantes para alguns usuários ou até mesmo para definir visualmente na interface o tamanho de um nó. O valor do cálculo de centralidade determina o nível de importância de um

Figura 18 – Exemplo de retorno com informações de um nó da GRN

```
[
  {
    "id": "EGX95353.1",
    "product": "37S ribosomal protein S12",
    "interpro": "IPR002594",
    "pfam": "PF01670",
    "go": "0005576",
    "gene3d": "P22669",
    "reactome": "R-HSA-114547",
    "panther": "PTHR34002",
    "uniprot": "014405",
    "ec_number": "3.2.1.4",
    "cazy": "GH12"
  }
]
```

Fonte: O autor

nó na rede e com essa informação, é possível alterar a rede, para que os nós mais importantes sejam maiores, oferecendo uma melhor experiência visual para o usuário.

Na Figura 19 temos um exemplo de retorno das medidas de centralidade de um determinado nó da GRN.

Figura 19 – Exemplo de retorno com medidas de centralidade um nó da GRN

```
[
  {
    "locus_tag": "CCM_00007",
    "degree": 0.0003193357815743254,
    "closeness": 0.00198959436261976,
    "betweenness": 0.0,
    "eigenvector": 0.011093787281242199,
    "harmonic": 15.24523809523809
  }
]
```

Fonte: O autor

#### 4.4.6 Criação de novo usuário - RF01 e RF02

Para que seja possível a utilização do sistema FunRegulation, será necessário que o usuário realize um cadastro em nosso sistema. Para realizar o cadastro o usuário deve informar seu primeiro e último nome, e-mail, senha, organização, país e estado, se este for nascido no Brasil. Na figura 20 podemos visualizar um exemplo de requisição para realizar o cadastro do usuário.



sua duração é de 30 segundos. O outro *token* é chamado de *refresh token*, e é armazenado nos cookies do navegador do usuário, sua duração é de 7 dias. A criação dos *tokens* é necessária para questões que envolvem segurança e gestão de sessões.

#### 4.4.8 Logout - RF04

Por questões de segurança foi criado o *end-point* de logout, pois no momento em que o usuário finalizar sua sessão no FunRegulation certas medidas devem ser tomadas. Quando o usuário desejar sair do sistema, o *token* que fica guardado nos cookies do navegador é excluído, tornando-o inválido, sendo necessário que o usuário realize novamente o login para obter um novo *token* para utilizar normalmente o sistema novamente.

#### 4.4.9 Solicitação de alteração de senha - RF05

Em caso de esquecimento da senha cadastrada pelo usuário, existe o *end-point* para o cadastro dessa solicitação. No sistema FunRegulation quando ocorrer de um usuário esquecer sua senha, será enviado um e-mail contendo um link personalizado exclusivamente para aquele usuário. Esse link foi construindo com um *token* gerado pelo próprio Django através de uma classe chamada *PasswordResetTokenGenerator* importada pelo módulo *django.contrib.auth.tokens*.

Para enviar a requisição, o usuário deve enviar preencher seu e-mail e receberá em sua caixa de entrada, conforme a Figura 22, o link possibilitando a troca da senha.

Figura 22 – E-mail com link para efetuar troca de senha



Fonte: O autor

#### 4.4.10 Confirmação de alteração de senha - RF05

Por fim, foi criado um *end-point* para realizar a validação do *token* de alteração da senha. Este *end-point* é responsável por verificar se o *token* gerado na URL da Figura 22 ainda é válido para aquele usuário. Esta validação de *token* é feita por um método próprio do Django,

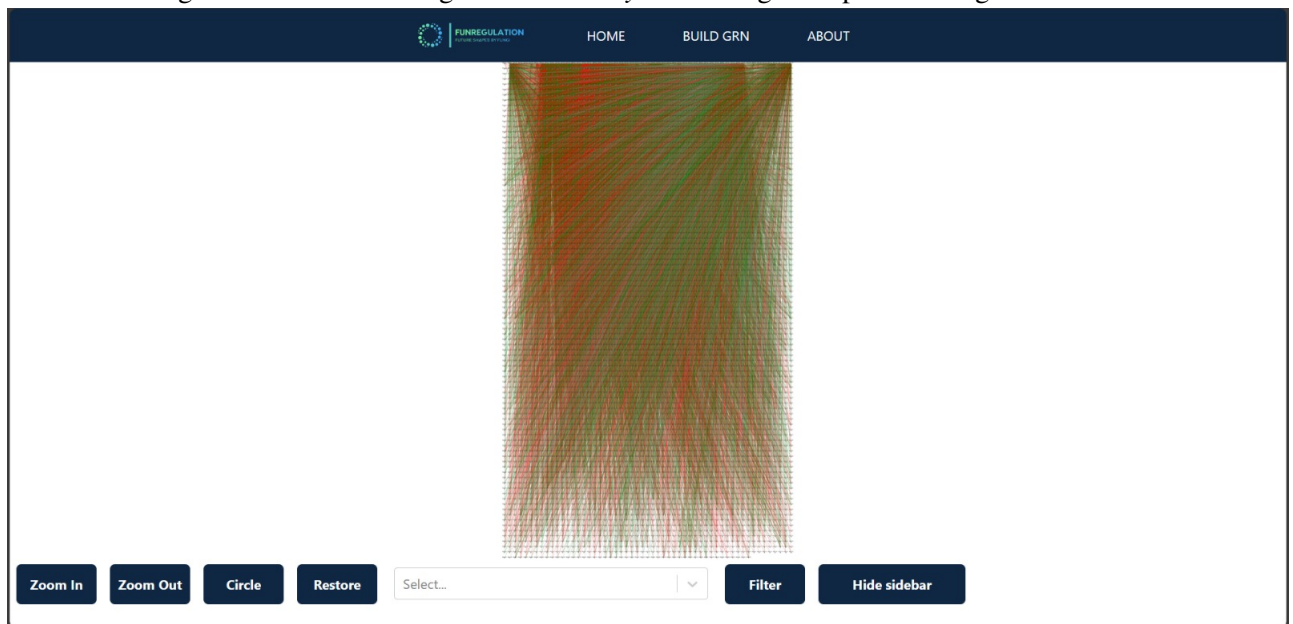
chamado *check token*, que recebe como argumento o ID do usuário e o *token* gerado na etapa de solicitação de alteração da senha.

Ambos os argumentos são enviados pela etapa anterior em forma de URL criptografada. Caso o *check token* consiga verificar a identidade do usuário e validar o *token*, a senha poderá ser trocada, caso contrário é necessário que o usuário solicite novamente a alteração de senha, para receber um novo e-mail.

## 5 VALIDAÇÃO

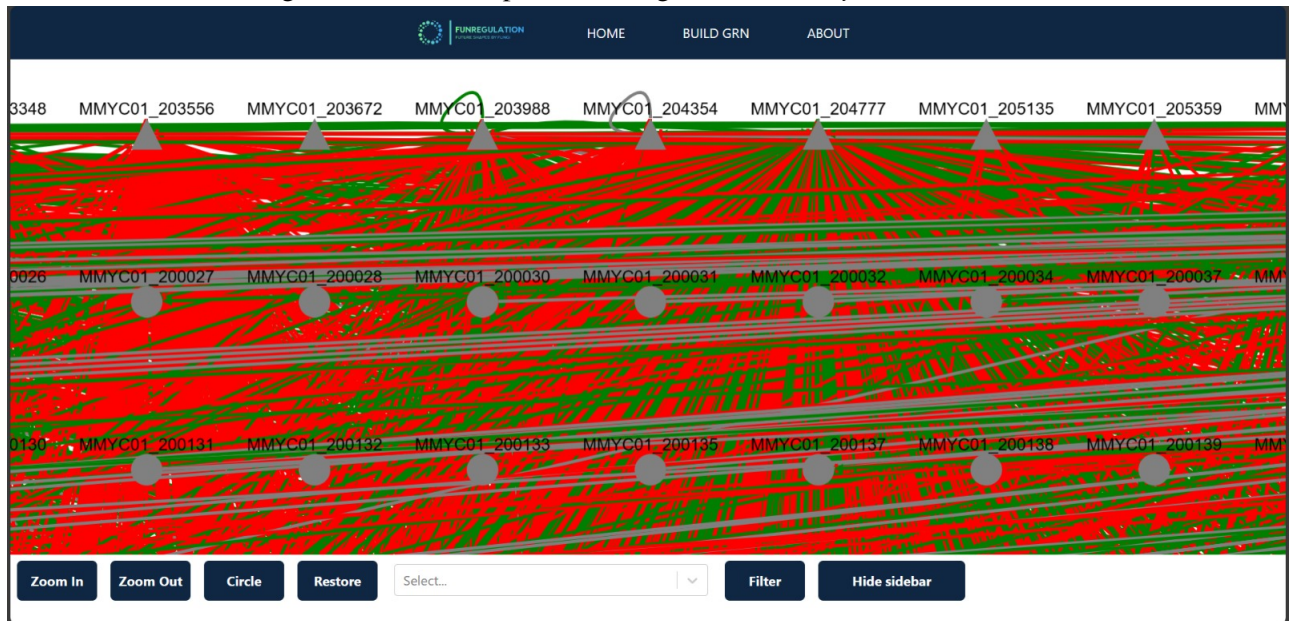
Para realizar a validação do sistema *FunRegulation*, construímos duas GRNs utilizando dois fungos de diferentes ordens. O intuito desta validação foi verificar se todo o *pipeline* era executado corretamente e atestar a geração da GRN, gerando os dados para serem enviados ao *front-end* para construir toda a rede de forma visual para o usuário. Dos dois fungos selecionados para a validação do *FunRegulation* selecionamos um fungo que possuía as matrizes de peso e o outro não, essa decisão foi tomada propositalmente para que as duas rotas do *pipeline* fossem testadas, uma com o *software* RSAT incluído no processo de geração de GRN e outro não, a fim de verificar o comportamento do sistema em ambos os casos.

Figura 23 – GRN do fungo *Madurella mycetomatis* gerada pela FunRegulation



Fonte: Samuel Santos Vieira

O primeiro fungo utilizado para gerar a GRN foi o *Madurella mycetomatis* da ordem *Sordariales*, cepa mm55 e de número de acesso GCA\_001275765.2. No *pipeline* de construção da GRN, não foi executado o RSAT, pois as matrizes de peso deste fungo ainda não são conhecidas. A GRN completa gerada pelo *FunRegulation* pode ser vista na Figura 23, a GRN ampliada de um trecho onde podemos ver mais claramente os TFs e os TGs do fungo, pode ser vista na Figura 24 e os dados obtidos como número de nós, quantidade de fatores de transcrição e outras estatísticas, podem ser vistas na Figura 25.

Figura 24 – GRN ampliada do fungo *Madurella mycetomatis*

Fonte: Samuel Santos Vieira

Figura 25 – Principais estatísticas da rede do fungo *Madurella mycetomatis*

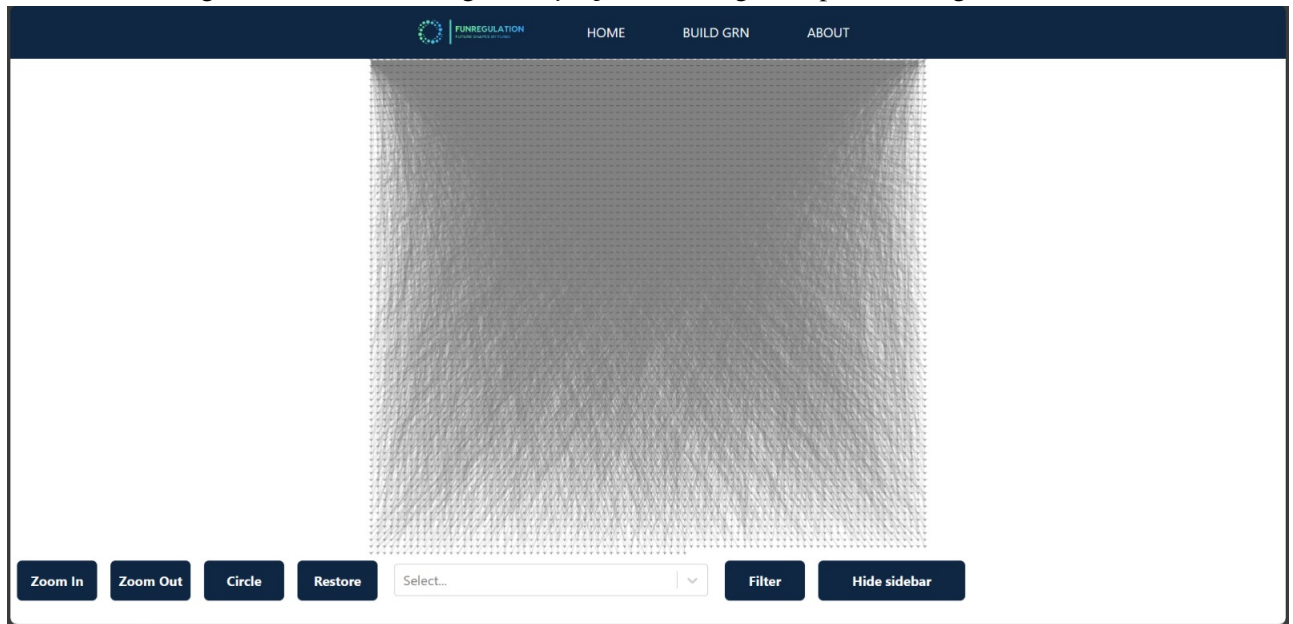
Informação	Quantidade
Interações	21.402
Nós	5.212
Autorregulações	28
Regulações positivas	9.181
Regulações negativas	8.373
Regulações desconhecidas	3.848

Fonte: O autor

O *Madurella mycetomatis* é o causador da doença micetoma, geralmente afeta crianças e jovens adultos em áreas rurais, causando um processo inflamatório severo na pele. É mais comum em trabalhadores rurais pois acredita-se que o fungo esteja presente no solo e infecte esses trabalhadores através de traumas na pele. A taxa de cura pode chegar até 90% porém o tratamento deve ser realizado rapidamente, pois caso a doença se espalhe pode causar graves lesões e até mesmo a morte (50).

O segundo fungo selecionado para construção da GRN foi o *Cordyceps militaris*, da ordem *Hypocreales*, cepa CM01 e com o número de acesso GCA\_000225605.1. Para a geração da GRN deste fungo foi possível incluir no *pipeline* o *software* RSAT para realizar a predição dos TFBSs, pois as matrizes de peso deste fungo encontram-se disponíveis no banco de dados

Figura 26 – GRN do fungo *Cordyceps militaris* gerada pela FunRegulation



Fonte: Samuel Santos Vieira

Cis-BP. A GRN que foi construída pode ser vista na Figura 26 e os dados estatísticos da rede podem ser vistas na Figura 27.

Conforme pode ser visto na Figura 26 todas as interações estão na cor cinza, diferentemente da Figura 23 onde as interações regulatórias possuem as cores vermelhas ou verdes. Isso ocorre pois o fungo que serve como modelo para transferimos conhecimento ao *Cordyceps militaris*, o fungo *Fusarium graminearum*, apesar de ser conhecida suas interações (13.153 nós e 39.459 interações regulatórias) ainda não existe a definição se essas interações são positivas (ativam) ou negativas (reprimem) para um determinado gene (51). Por esse motivo não é possível realizar a transferência dessa informação ao fungo alvo *Cordyceps militaris*.

O *Cordyceps militaris* é um fungo com uma vasta gama de estudos, sendo bastante utilizado na medicina tradicional chinesa nos últimos 300 anos (52). Pesquisas recentes apontam que o fungo possui diversas propriedades benéficas à saúde do ser humano, como propriedades antioxidantes, anticancerígenas, anti bacterianas, anti inflamatórias, neuroprotetoras e diversas outras (52, 53, 54). Estudos também apontam que o *Cordyceps militaris* pode melhorar a performance em exercícios de altas intensidades como no estudo (55).

Para concluirmos a nossa validação, devemos responder o principal critério de avaliação definido na metodologia de pesquisa DSR: “O artefato consegue construir uma GRN?”. Podemos responder essa avaliação de maneira positiva pois, conforme visto nas imagens anterior-

Figura 27 – Principais estatísticas da rede do fungo *Cordyceps militaris*

Informação	Quantidade
Interações	14.847
Nós	6.345
Autorregulações	0
Regulações positivas	0
Regulações negativas	0
Regulações desconhecidas	14.847

Fonte: O autor

res que mostram as GRNs dos fungos *Madurella mycetomatis* e *Cordyceps militaris* juntamente com suas respectivas informações, demonstramos que o *FunRegulation* foi capaz de gerar GRNs para fungos de diferentes características e modelos, conjuntamente com a utilização de duas rotas distintas do *pipeline* para a construção destas redes.

## 6 CONSIDERAÇÕES FINAIS

A arquitetura *Web Back-end* para construção de GRNs de fungos desenvolvida neste trabalho, parte do sistema *Web* chamado *FunRegulation*, contribui para o processo de entendimento das relações entre os genes dos fungos, de maneira significativa. O processo de construção de GRNs, além de ter a possibilidade de ser disponibilizado na *Web* via API, também está inteiramente unificado, restando apenas a integração ao *Front-end*, para posteriormente ser distribuído para toda a comunidade científica interessada em compreender melhor os fungos.

Outro aspecto importante de ser citado neste trabalho em relação à arquitetura, é sua capacidade de receber futuras atualizações e aprimoramentos como, por exemplo, a introdução de novas ferramentas, sem a necessidade de extensas alterações estruturais. Essa facilidade ocorre pois com o uso do *Celery* e do *RabbitMQ*, todas as tarefas do sistema (chamada de ferramentas, consultas em banco de dados) foram modularizadas e organizadas em tarefas assíncronas, deixando a solução preparada para suportar novos módulos e melhorias gerais.

A implementação da arquitetura *Web Back-end* do *FunRegulation* foi testada e validada para a geração de GRNs utilizando dois fungos de diferentes ordens, sendo o *Madurella mycetomatis* e o *Cordyceps militaris*. Ambas GRNs foram salvas no banco de dados e enviadas no formato JSON, para que o *Front-end* pudesse construir suas respectivas GRNs no formato de grafos, ou seja, o formato que será visível para os usuários que utilizarem o sistema futuramente.

Conforme definido nos objetivos específicos deste trabalho, dos 4 objetivos designados para esta pesquisa 3 foram alcançados em sua totalidade. Conseguimos atingir o objetivo de identificação das etapas e ferramentas para compor o *pipeline* de geração das GRNs, através do *proteinOrtho*, *RSAT* e medidas de centralidade. Também foram alcançados os objetivos de definição de modelo para gerenciamento e escalonamento das etapas de construção das GRNs, através da arquitetura de *software* que construímos com o *Django*, *RabbitMQ* e *Celery* e por fim, o objetivo de construção de duas GRNs de fungos distintos para realizarmos a validação final deste trabalho, com a geração das GRNs mostradas na seção de validação.

O único objetivo que não foi alcançado em sua totalidade, se refere à definição do modelo responsável por integrar as ferramentas de terceiros como o *ProteinOrtho* e *RSAT*. Ambos

os *softwares* foram instalados de maneira local, por questões de praticidade para a realização de testes. Porém, existe uma maneira mais apropriada para a execução destas ferramentas, que é através da utilização de um *software* para criação de *containers*, como um *Docker*. Com a utilização de um *Docker*, o processo de implementação e integração do *FunRegulation* será bastante facilitado, pois não será necessário realizar novamente a instalação das ferramentas individualmente, bastando replicar o *container* sempre que necessário.

Como trabalhos futuros uma série de melhorias podem ser implementadas, visando obter um *software* mais modularizado e seguro. O primeiro ponto que deve ser aprimorado é a questão da integração das ferramentas de terceiros. Os *softwares* de terceiros como ProteinOrtho e RSAT foram instalados localmente, com o intuito de agilizar o processo de desenvolvimento, porém como o FunRegulation será disponibilizado para a Web, a utilização desses *softwares* através de *containers* Docker é necessária.

Com a utilização de um container toda a aplicação pode se manter isolada sem a dependência de uma máquina específica, conseqüentemente as futuras manutenções e integrações ficarão facilitadas. Com a aplicação isolada em um Docker por exemplo, será possível reutilizar um *container* em outras máquinas/servidores sem a dependência de um ambiente pré-configurado com as bibliotecas e/ou ferramentas previamente instaladas.

Outro trabalho futuro que deve ser realizado é a integração deste trabalho, com o *Front-end*. Ambas as partes já foram desenvolvidas e testadas através de arquivos JSON que foram gerados pelo *Back-end* e enviados ao *Front-end*, porém essa comunicação precisa ser automatizada para disponibilização da aplicação de maneira completa para a comunidade científica. Somente com a disponibilização da integração completa do FunRegulation para a comunidade científica será possível respondermos à avaliação 2 definida na metodologia DSR na Figura 9: “O artefato consegue contribuir positivamente com a comunidade científica?”.

## REFERÊNCIAS

- 1 BAYAT, A. Science, medicine, and the future: Bioinformatics. **BMJ**, England, v. 324, n. 7344, p. 1018–1022, abr. 2002.
- 2 JACKSON, C. A.; CASTRO, D. M.; SALDI, G.-A.; BONNEAU, R.; GRESHAM, D. Gene regulatory network reconstruction using single-cell rna sequencing of barcoded genotypes in diverse environments. **eLife**, eLife Sciences Publications, Ltd, v. 9, p. e51254, jan 2020. ISSN 2050-084X. Disponível em: <<https://doi.org/10.7554/eLife.51254>>.
- 3 LEIPZIG, J. A review of bioinformatic pipeline frameworks. **Briefings in Bioinformatics**, v. 18, n. 3, p. 530–536, 03 2016. ISSN 1467-5463. Disponível em: <<https://doi.org/10.1093/bib/bbw020>>.
- 4 KARLEBACH, G.; SHAMIR, R. Modelling and analysis of gene regulatory networks. **Nat Rev Mol Cell Biol**, England, v. 9, n. 10, p. 770–780, set. 2008.
- 5 LENZ, A. R.; GALÁN-VÁSQUEZ, E.; BALBINOT, E.; ABREU, F. P. de; OLIVEIRA, N. Souza de; ROSA, L. O. da; SILVA, S. de Avila e; CAMASSOLA, M.; DILLON, A. J. P.; PEREZ-RUEDA, E. Gene regulatory networks of penicillium echinulatum 2hh and penicillium oxalicum 114-2 inferred by a computational biology approach. **Frontiers in Microbiology**, v. 11, 2020. ISSN 1664-302X. Disponível em: <<https://www.frontiersin.org/articles/10.3389/fmicb.2020.588263>>.
- 6 COSTA, D. A. Genômica comparativa de leveduras de interesse biotecnológico. Universidade Federal de Viçosa, 2015.
- 7 SAYERS, E. W.; BECK, J.; BOLTON, E. E.; BOUREXIS, D.; BRISTER, J. R.; CANESE, K.; COMEAU, D. C.; FUNK, K.; KIM, S.; KLIMKE, W.; MARCHLER-BAUER, A.; LANDRUM, M.; LATHROP, S.; LU, Z.; MADDEN, T. L.; O'LEARY, N.; PHAN, L.; RANGWALA, S. H.; SCHNEIDER, V. A.; SKRIPCHENKO, Y.; WANG, J.; YE, J.; TRAWICK, B. W.; PRUITT, K. D.; SHERRY, S. T. Database resources of the National Center for Biotechnology Information. **Nucleic Acids Research**, v. 49, n. D1, p. D10–D17, 10 2020. ISSN 0305-1048. Disponível em: <<https://doi.org/10.1093/nar/gkaa892>>.
- 8 WYK, R. van; BILJON, R. van; BIRKHOLTZ, L.-M. MALBoost: a web-based application for gene regulatory network analysis in Plasmodium falciparum. **Malaria Journal**, v. 20, n. 1, p. 317, jul. 2021. ISSN 1475-2875. Disponível em: <<https://doi.org/10.1186/s12936-021-03848-2>>.
- 9 CANO, R.; LENZ, A. R.; GALAN-VASQUEZ, E.; RAMIREZ-PRADO, J. H.; PEREZ-RUEDA, E. Gene regulatory network inference and gene module regulating virulence in fusarium oxysporum. **Front. Microbiol.**, Frontiers Media SA, v. 13, jun. 2022.
- 10 LENZ, A. R. **GitHub - G2BC/funregulation: FunRegulation: Gene regulatory networks (GRN) of Penicillium echinulatum 2HH and Penicillium oxalicum 114-2 inferred by a computational biology approach — github.com.** <<https://github.com/G2BC/funregulation>>. [Accessed 11-12-2023].

- 11 SETIC-UFSC. **Nem bicho, nem planta: o que nos ensina o conhecimento dos fungos — noticias.ufsc.br.** 2022. <<https://noticias.ufsc.br/2022/11/nem-bicho-nem-planta-o-que-nos-ensina-o-conhecimento-dos-fungos>>. [Accessed 20-12-2023].
- 12 HYDE, K. D.; XU, J.; RAPIOR, S.; JEEWON, R.; LUMYONG, S.; NIEGO, A. G. T.; ABEYWICKRAMA, P. D.; ALUTHMUHANDIRAM, J. V. S.; BRAHAMANAGE, R. S.; BROOKS, S.; CHAIYASEN, A.; CHETHANA, K. W. T.; CHOMNUNTI, P.; CHEPKIRUI, C.; CHUANKID, B.; SILVA, N. I. de; DOILOM, M.; FAULDS, C.; GENTEKAKI, E.; GOPALAN, V.; KAKUMYAN, P.; HARISHCHANDRA, D.; HEMACHANDRAN, H.; HONGSANAN, S.; KARUNARATHNA, A.; KARUNARATHNA, S. C.; KHAN, S.; KUMLA, J.; JAYAWARDENA, R. S.; LIU, J.-K.; LIU, N.; LUANGHARN, T.; MACABEO, A. P. G.; MARASINGHE, D. S.; MEEKS, D.; MORTIMER, P. E.; MUELLER, P.; NADIR, S.; NATARAJA, K. N.; NONTACHAIYAPOOM, S.; O'BRIEN, M.; PENKHRUE, W.; PHUKHAMSAKDA, C.; RAMANAN, U. S.; RATHNAYAKA, A. R.; SADABA, R. B.; SANDARGO, B.; SAMARAKOON, B. C.; TENNAKOON, D. S.; SIVA, R.; SRIPROM, W.; SURYANARAYANAN, T. S.; SUJARIT, K.; SUWANNARACH, N.; SUWUNWONG, T.; THONGBAI, B.; THONGKLANG, N.; WEI, D.; WIJESINGHE, S. N.; WINISKI, J.; YAN, J.; YASANTHIKA, E.; STADLER, M. The amazing potential of fungi: 50 ways we can exploit fungi industrially. **Fungal Diversity**, v. 97, n. 1, p. 1–136, Jul 2019. ISSN 1878-9129. Disponível em: <<https://doi.org/10.1007/s13225-019-00430-9>>.
- 13 MERCATELLI, D.; SCALAMBRA, L.; TRIBOLI, L.; RAY, F.; GIORGI, F. M. Gene regulatory network inference resources: A practical overview. **Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms**, v. 1863, n. 6, p. 194430, jun. 2020. ISSN 1874-9399. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1874939919300410>>.
- 14 GABALDÓN, T.; KOONIN, E. V. Functional and evolutionary implications of gene orthology. **Nat. Rev. Genet.**, Springer Science and Business Media LLC, v. 14, n. 5, p. 360–366, maio 2013.
- 15 JENSEN, R. A. Orthologs and paralogs - we need to get it right. **Genome Biology**, v. 2, n. 8, p. interactions1002.1, Aug 2001. ISSN 1474-760X. Disponível em: <<https://doi.org/10.1186/gb-2001-2-8-interactions1002>>.
- 16 NICHIO, B. T.; MARCHAUKOSKI, J. N.; RAITTZ, R. T. New tools in orthology analysis: a brief review of promising perspectives. **Frontiers in genetics**, Frontiers Media SA, v. 8, p. 165, 2017.
- 17 Zogopoulos; Vasileios; L.; SPAHO, K.; NTOUKA, C.; Lappas; Gerasimos; A.; KYRANIS, I.; Bagos; Pantelis; G.; Spandidos; Demetrios; A.; MICHALOPOULOS, I. TFBSPred: A functional transcription factor binding site prediction webtool for humans and mice. **Int J Epigen**, v. 1, n. 4, p. 9, nov. 2021.
- 18 KHAMIS, A. M.; MOTWALLI, O.; OLIVA, R.; JANKOVIC, B. R.; MEDVEDEVA, Y. A.; ASHOOR, H.; ESSACK, M.; GAO, X.; BAJIC, V. B. A novel method for improved accuracy of transcription factor binding site prediction. **Nucleic Acids Research**, v. 46, n. 12, p. e72–e72, 04 2018. ISSN 0305-1048. Disponível em: <<https://doi.org/10.1093/nar/gky237>>.
- 19 ZENG, Y.; GONG, M.; LIN, M.; DONGRUI, G.; ZHANG, Y. A review about transcription factor binding sites prediction based on deep learning. **IEEE Access**, v. 8, p. 219256–219274, 01 2020.

- 20 LECHNER, M.; FINDEISS, S.; STEINER, L.; MARZ, M.; STADLER, P. F.; PROHASKA, S. J. Proteinortho: Detection of (co-)orthologs in large-scale analysis. **BMC Bioinformatics**, v. 12, n. 1, p. 124, Apr 2011. ISSN 1471-2105. Disponível em: <<https://doi.org/10.1186/1471-2105-12-124>>.
- 21 PAUL KLEMM. **PHD proteinortho GitLab**. 2022. Disponível em: <[https://gitlab.com/paulklemm\\_PHD/proteinortho](https://gitlab.com/paulklemm_PHD/proteinortho)>. Acesso em: 01 de janeiro 2022.
- 22 CIS-BP Database: Catalog of Inferred Sequence Binding Preferences — [cisbp.cabr.utoronto.ca](http://cisbp.cabr.utoronto.ca). <<http://cisbp.cabr.utoronto.ca/>>. [Accessed 20-11-2023].
- 23 NGUYEN, N.; CONTRERAS-MOREIRA, B.; CASTRO-MONDRAGON, J. A.; SANTANA-GARCIA, W.; OSSIO, R.; ROBLES-ESPINOZA, C.; BAHIN, M.; COLLOMBET, S.; VINCENS, P.; THIEFFRY, D.; VAN HELDEN, J.; MEDINA-RIVERA, A.; THOMAS-CHOLLIER, M. RSAT 2018: regulatory sequence analysis tools 20th anniversary. **Nucleic Acids Research**, v. 46, n. W1, p. W209–W214, 05 2018. ISSN 0305-1048. Disponível em: <<https://doi.org/10.1093/nar/gky317>>.
- 24 MEDINA-RIVERA, A.; DEFRANCE, M.; SAND, O.; HERRMANN, C.; CASTRO-MONDRAGON, J. A.; DELERCE, J.; JAEGER, S.; BLANCHET, C.; VINCENS, P.; CARON, C.; STAINES, D. M.; CONTRERAS-MOREIRA, B.; ARTUFEL, M.; CHARBONNIER-KHAMVONGSA, L.; HERNANDEZ, C.; THIEFFRY, D.; THOMAS-CHOLLIER, M.; HELDEN, J. van. RSAT 2015: Regulatory Sequence Analysis Tools. **Nucleic Acids Research**, v. 43, n. W1, p. W50–W56, 04 2015. ISSN 0305-1048. Disponível em: <<https://doi.org/10.1093/nar/gkv362>>.
- 25 BASSI, S. A primer on python for life science researchers. **PLoS computational biology**, Public Library of Science San Francisco, USA, v. 3, n. 11, p. e199, 2007.
- 26 SAABITH, A.; FAREEZ, M.; VINOThRAJ, T. Python current trend applications-an overview. **International Journal of Advance Engineering and Research Development**, v. 6, n. 10, 2019.
- 27 PYTHON. **Welcome to Python.org**. 2022. Disponível em: <<https://www.python.org/>>. Acesso em: 01 de janeiro 2022.
- 28 B, V. **artima - The Making of Python** — [artima.com](https://www.artima.com/articles/the-making-of-python). 2003. <<https://www.artima.com/articles/the-making-of-python>>. [Accessed 20-12-2023].
- 29 GAUTHIER, J.; VINCENT, A. T.; CHARETTE, S. J.; DEROME, N. A brief history of bioinformatics. **Briefings in Bioinformatics**, v. 20, n. 6, p. 1981–1996, 08 2018. ISSN 1477-4054. Disponível em: <<https://doi.org/10.1093/bib/bby063>>.
- 30 EKMEKCI, B.; MCANANY, C. E.; MURA, C. An introduction to programming for bioscientists: A python-based primer. **PLoS Comput. Biol.**, Public Library of Science (PLoS), v. 12, n. 6, p. e1004867, jun. 2016.
- 31 COCK, P. J. A.; ANTAO, T.; CHANG, J. T.; CHAPMAN, B. A.; COX, C. J.; DALKE, A.; FRIEDBERG, I.; HAMELRYCK, T.; KAUFF, F.; WILCZYNSKI, B.; HOON, M. J. L. de. Biopython: freely available Python tools for computational molecular biology and bioinformatics. **Bioinformatics**, v. 25, n. 11, p. 1422–1423, 03 2009. ISSN 1367-4803. Disponível em: <<https://doi.org/10.1093/bioinformatics/btp163>>.

- 32 POSTGRESQL: About — postgresql.org. <<https://www.postgresql.org/about/>>. [Accessed 13-11-2023].
- 33 OLUWATOSIN, H. S. Client-server model. **IOSR Journal of Computer Engineering**, IOSR Journals, v. 16, n. 1, p. 67–71, 2014.
- 34 CELERY - Distributed Task Queue; Celery 5.3.5 documentation — docs.celeryq.dev. <<https://docs.celeryq.dev/en/stable/>>. [Accessed 13-11-2023].
- 35 ZHENG, M.; ZHANG, S.; ZHANG, Y.; HU, B. Construct food safety traceability system for people's health under the internet of things and big data. **IEEE Access**, v. 9, p. 70571–70583, 2021.
- 36 ROY, G. M. **RabbitMQ in Depth**. New York, NY: Manning Publications, 2016.
- 37 ARASA, D.; MEHARUNNISA, S. Development of android application for device to device communication in iot using rabbit mq broker. **Int J Adv Res Innovation**, v. 7, n. 3, p. 251–252, 2019.
- 38 BEN GUEBILA, M.; LOPES-RAMOS, C. M.; WEIGHILL, D.; SONAWANE, A.; BURKHOLZ, R.; SHAMSAEI, B.; PLATIG, J.; GLASS, K.; KUIJER, M.; QUACKENBUSH, J. GRAND: a database of gene regulatory network models across human conditions. **Nucleic Acids Research**, v. 50, n. D1, p. D610–D621, set. 2021. ISSN 0305-1048. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8728257/>>.
- 39 DING, J.; BLENCOWE, M.; NGHIEM, T.; HA, S.-M.; CHEN, Y.-W.; LI, G.; YANG, X. Mergeomics 2.0: a web server for multi-omics data integration to elucidate disease networks and predict therapeutics. **Nucleic Acids Research**, v. 49, n. W1, p. W375–W387, jul. 2021. ISSN 1362-4962.
- 40 PIMENTEL, M.; FILIPPO, D.; SANTOS, T. M. D. Design science research: pesquisa científica atrelada ao design de artefatos. **Revista de Educacao a Distancia e Elearning**, Vol. 3 N.º 1, 2020. Disponível em: <[https://revistas.rcaap.pt/lead\\_read/article/view/21898](https://revistas.rcaap.pt/lead_read/article/view/21898)>.
- 41 WHAT is Scrum? — scrum.org. <<https://www.scrum.org/learning-series/what-is-scrum>>. [Accessed 11-12-2023].
- 42 PEEK, S. What is agile scrum methodology. **Luettavissa: https://www.businessnewsdaily.com/4987-what-is-agile-scrum-methodology.html. Luettu**, v. 22, p. 2022, 2022.
- 43 PERMANA, P. A. G. Scrum method implementation in a software development project management. **International Journal of Advanced Computer Science and Applications**, Science and Information (SAI) Organization Limited, v. 6, n. 9, p. 198–204, 2015.
- 44 AGARWAL, S.; AGARWAL, A. Uses, advantages and opportunities of kanban methods in mechanical engineering and product manufacturing. **Int. J. Sci. Res. Publ. (IJSRP)**, International Journal of Scientific and Research Publications (IJSRP), v. 10, n. 1, p. 9704, jan. 2020.
- 45 MANAGE Your Team2019;s Projects From Anywhere | Trello — trello.com. <<https://trello.com/en>>. [Accessed 20-11-2023].
- 46 GITHUB: Let's build from here — github.com. <<https://github.com/>>. [Accessed 20-11-2023].

- 47 COSENTINO, V.; IZQUIERDO, J. L. C.; CABOT, J. A systematic mapping study of software development with github. **Ieee access**, IEEE, v. 5, p. 7173–7192, 2017.
- 48 FUNREGULATIONAPI — github.com. 2023. <<https://github.com/G2BC/FunRegulationAPI/tree/main/funRegulationBackCode>>. [Accessed 16-11-2023].
- 49 HELDEN, B. C. M. Jacques van. **Installing RSAT on a Unix operating system — rsa-tools.github.io**. <[https://rsa-tools.github.io/installing-RSAT/unix-install-rsat/installing\\_RSAT\\_procedure.html](https://rsa-tools.github.io/installing-RSAT/unix-install-rsat/installing_RSAT_procedure.html)>. [Accessed 21-11-2023].
- 50 ELKHEIR, L. Y. M.; HAROUN, R.; MOHAMED, M. A.; FAHAL, A. H. Madurella mycetomatis causing eumycetoma medical treatment: The challenges and prospects. **PLoS Negl. Trop. Dis.**, Public Library of Science (PLoS), v. 14, n. 8, p. e0008307, ago. 2020.
- 51 GUO, L.; HAN, L.; YANG, L.; ZENG, H.; FAN, D.; ZHU, Y.; FENG, Y.; WANG, G.; PENG, C.; JIANG, X.; ZHOU, D.; NI, P.; LIANG, C.; LIU, L.; WANG, J.; MAO, C.; FANG, X.; PENG, M.; HUANG, J. Genome and transcriptome analysis of the fungal pathogen fusarium oxysporum f. sp. cubense causing banana vascular wilt disease. **PLoS One**, Public Library of Science (PLoS), v. 9, n. 4, p. e95543, abr. 2014.
- 52 ASHRAF, S. A.; ELKHALIFA, A. E. O.; SIDDIQUI, A. J.; PATEL, M.; AWADELKA-REEM, A. M.; SNOUSSI, M.; ASHRAF, M. S.; ADNAN, M.; HADI, S. Cordycepin for health and wellbeing: A potent bioactive metabolite of an entomopathogenic medicinal fungus cordyceps with its nutraceutical and therapeutic potential. **Molecules**, v. 25, n. 12, 2020. ISSN 1420-3049. Disponível em: <<https://www.mdpi.com/1420-3049/25/12/2735>>.
- 53 TULI, H. S.; SANDHU, S. S.; SHARMA, A. K. Pharmacological and therapeutic potential of cordyceps with special reference to cordycepin. **3 Biotech**, Springer Science and Business Media LLC, v. 4, n. 1, p. 1–12, fev. 2014.
- 54 SHWETA; ABDULLAH, S.; KOMAL; KUMAR, A. A brief review on the medicinal uses of cordyceps militaris. **Pharmacological Research - Modern Chinese Medicine**, v. 7, p. 100228, 2023. ISSN 2667-1425. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2667142523000143>>.
- 55 HIRSCH, K. R.; SMITH-RYAN, A. E.; ROELOFS, E. J.; TREXLER, E. T.; MOCK, M. G. *Cordyceps militaris* improves tolerance to high-intensity exercise after acute and chronic supplementation. **J. Diet. Suppl.**, Informa UK Limited, v. 14, n. 1, p. 42–53, jan. 2017.

## **APÊNDICES**

## APÊNDICE A – Revisão sistemática

# **Uma Revisão de Literatura Sobre *Plataformas Web Para Inferência e Visualização de Redes Regulatórias de Genes***

**Gabriel Antônio Alves de Amorim**

<sup>1</sup>Departamento de Ciências Exatas e da Terra, Campus I  
Universidade do Estado da Bahia (UNEB)  
Salvador, Bahia, Brasil.

[gabrielfai70@gmail.com](mailto:gabrielfai70@gmail.com)

**Resumo.** *A bioinformática é uma das principais áreas responsáveis pelo desenvolvimento atual das ciências da vida. À medida que os avanços tecnológicos proporcionaram importantes descobertas na área da biologia molecular, ampliou-se também a complexidade e os recursos necessários para o entendimento e, conseqüentemente, para o desenvolvimento de mais conhecimento. Esta revisão da literatura demonstra detalhadamente, a metodologia utilizada para busca e análise de abordagens para desenvolvimento de aplicações Web que implementam a construção e manipulação de redes regulatórias de genes, tendo como principal objetivo responder questões de pesquisa e compreender o estado da arte sobre redes regulatórias de genes.*

**Abstract.** *Bioinformatic is today, one of the areas responsible for the actual development of biological sciences. As long as the technological advancements provided important discoveries in the molecular biology, widened the complexity and the necessary resources, for the understanding and consequently for the development of more knowledge. This lecture's revision shows in detail the methodology in the search and analysis of approaches about gene regulatory networks on the web. Having as main objective answer research questions and understand the state of the art about gene regulatory networks.*

## **1. Introdução**

O avanço tecnológico vem proporcionando ao longo do tempo, diversas transformações nos mais variados campos do conhecimento humano. Uma das áreas mais afetadas de maneira positiva, sem dúvida, é a área das ciências da vida, em especial a genômica. Conforme a capacidade computacional e o poder de processamento de grande quantidade de dados vem sendo expandidos, amplia-se também o leque de estudos e possibilidades que antes seriam impossíveis de serem realizadas, devido à carência de tecnologia para servir de apoio nas construções dessas pesquisas.

A bioinformática é uma área onde são desenvolvidos métodos e softwares para analisar dados biológicos, a fim de entender o funcionamento dos seres vivos. Um dos campos de estudo é o das redes regulatórias de genes, onde são identificadas proteínas ou conjuntos de proteínas, na maioria dos casos essas proteínas são fatores de transcrição, que agem para estimular ou inibir a ativação de determinados genes. O entendimento dessas relações pode ajudar a aprimorar o conhecimento sobre diversos organismos, como

aponta [Aj et al. 2017] ou o desenvolvimento de remédios/tratamentos conforme mostra o artigo [Nguyen et al. 2020].

Este trabalho visa realizar um levantamento de tecnologias e pesquisas que tenham como foco a apresentação de ferramentas ou modelos para o desenvolvimento de Redes Regulatórias de Genes, buscando assim, identificar lacunas de pesquisa que permitam o desenvolvimento de propostas para um Trabalho de Conclusão de Curso – TCC.

Para ajudar a compreender melhor o tema, foram elaboradas algumas questões de pesquisa, que são:

- Quais arquiteturas utilizadas em plataformas web para visualização e manipulação de uma Gene Regulatory Network (GRN)?
- Quais são as tecnologias utilizadas para construção de plataformas web para visualização e manipulação de uma GRN?
- Como é feita a visualização da GRN em plataformas web para visualização e manipulação de uma GRN?
- Quais as funcionalidades de plataformas web para visualização e manipulação de uma GRN?
- Quais os desafios apresentados na construção de plataformas web para visualização e manipulação de uma GRN?

## 2. Relato da Revisão de Literatura

Para a realização da busca dos artigos, foi definido um intervalo de publicações dos últimos dez anos. Sendo assim, foram pesquisados artigos de 2012 a 2022. Sendo assim, foram pesquisados artigos de 2012 até 2022, nos repositórios ACM Digital Library, IEEE Digital Library, PMC, PubMed, ScienceDirect e SpringerLink. Eles foram selecionados pelo fato do tema da revisão estar ligado à área de bioinformática.

A definição dos critérios de inclusão e exclusão foi feita de acordo com as respostas que precisavam ser encontradas para as questões de pesquisa estabelecidas, com a intenção de apenas fazer a seleção dos artigos que pudessem respondê-las. Os critérios podem ser vistos na Tabela 1, que se encontra a seguir.

Identificador	Critério
I1	Recomendado pelo orientador.
I2	Aborda a construção, visualização ou inferência de redes regulatórias de genes.
E1	Artigo duplicado.
E2	Sem implementação de tecnologias.
E3	Não está relacionado a redes regulatórias de genes.
E4	Sem visualização de redes regulatórias de genes.
E5	Sem criação de quaisquer interfaces.

**Tabela 1. Lista dos critérios de inclusão (I) e exclusão (E) definidos para a fase de seleção da revisão sistemática.**

A string de busca definida neste trabalho, é composta por 4 (quatro) palavras-chaves, são elas: "gene regulatory network", "web server", "web interface" e "network vi-

sualization". A palavra-chave "gene regulatory network" foi escolhida por se tratar exatamente do tema de pesquisa ao qual está sendo feita a revisão sistemática, "web server" e "web interface" foram selecionadas para identificar como as redes reguladoras de gene são implantadas no contexto web e por fim a palavra-chave "network visualization" entender como essas redes, podem ser construídas e apresentadas para o usuário, buscando ferramentas ou métodos para apresentação de grandes redes de genes. Por fim, a string completa pode ser vista abaixo:

"gene regulatory network"AND ("web server"OR "web interface"OR "network visualization")

### **2.1. Busca no PMC**

No PMC, também foi utilizada a busca avançada. No query builder, foram selecionados os campos de título e resumo, com a condição "OR" entre eles, e a string de busca definida para a pesquisa foi utilizada para ambos. Após realizar a busca, foi selecionada no filtro de data de publicação a opção de dez anos.

### **2.2. Busca no PubMed**

Para a busca no PubMed, foi selecionada a opção "Advanced", abaixo da barra de pesquisa. No alvo de busca, foi escolhida a opção "text/abstract", e na caixa de texto ao lado foi inserida a string de busca. Por fim, clicou-se no botão "Add", para adicionar nossos termos de busca à query, e no botão "Search", para efetuar a pesquisa. Na tela de resultados, foi aplicado o filtro de publicações com no máximo dez anos.

### **2.3. Busca no ScienceDirect**

No ScienceDirect, foi selecionada a opção de busca avançada e em seguida exibidos todos os campos de busca. No campo de título, resumo e palavras-chave especificadas pelo autor, foi inserida a string de busca, e no campo de ano, o intervalo de 2012 a 2022.

### **2.4. Busca no SpringerLink**

No SpringerLink, a busca foi feita com a opção de busca avançada. Nela, a pesquisa foi limitada pelo título dos artigos e intervalo de anos de publicação entre 2012 a 2022. Como o resultado retornado continha somente artigos no idioma inglês, nenhum outro filtro precisou ser aplicado.

### **2.5. Resultados Parciais**

No total, foram retornados 50 artigos nas sessões de busca efetuadas. Desses artigos, 6 (11,5%) foram do ACM, 2 (3,8%) do IEEE Digital Library, 16 (30,8%) do PMC, 17 (32,7%) no PubMed, 2 (3,8%) no ScienceDirect, 7 (13,5%) no SpringerLink e 2 (3,8%) recomendado pelo orientador. A Figura 1, que está abaixo, apresenta o gráfico dos artigos encontrados.

Para fazer a seleção, foi feita a leitura do título, palavras-chave e resumo de cada um dos artigos, e em seguida realizada a avaliação considerando os critérios de seleção estabelecidos no protocolo da revisão sistemática. Do total de artigos, 23 (45,1%) foram rejeitados, 5 (9,8%) já haviam sido encontrados em sessões anteriores, e 23 (45,1%) foram



**Figura 1. Gráfico dos artigos encontrados em relação aos repositórios acadêmicos.**



**Figura 2. Gráfico da relação de artigos totais mediante sua classificação na fase de seleção.**

selecionados. A Figura 2 apresenta o gráfico dos artigos em relação à classificação após a fase de seleção.

Entre os 28 artigos rejeitados e duplicados, 5 (13,9%) se adequaram ao critério E1 - artigo duplicado, 15 (41,7%) ao critério E2 - sem implementação de tecnologias, 8 (22,2%) ao critério E3 - não está relacionado a redes regulatórias de genes, 3 (8,3%) ao critério E4 - sem visualização de redes regulatórias de genes e 5 (13,9%) ao critério E5 - sem criação de quaisquer interfaces. Como um artigo pode se adequar a mais de um critério de exclusão, a soma dos artigos que se adequam aos critérios é maior do que o

total de artigos rejeitados e duplicados. A Figura 3 mostra a relação entre o número de artigos e cada critério de exclusão.



**Figura 3. Gráfico de artigos rejeitados em relação aos critérios de exclusão.**

Dos 24 artigos aceitos, 17 (71%) foram encontrados em sessões no Google Scholar, 4 (17%) no SpringerLink e 3 (12%) adicionados manualmente. Como dito anteriormente, a quantidade de artigos aceitos no SpringerLink e Scielo levantou a necessidade de acrescentar o Google Scholar aos repositórios escolhidos. A Figura 4 mostra a quantidade de artigos selecionados em cada um dos repositórios acadêmicos.



**Figura 4. Gráfico de artigos aceitos em relação ao repositório acadêmico onde foi encontrado.**

### 3. Análise Qualitativa dos Resultados

Para a etapa de extração, foram observados alguns pontos dos artigos que ajudassem a responder às questões de pesquisa definidas para esta revisão sistemática. Após a leitura de cada um deles foi realizada uma reavaliação, para saber se realmente se encaixavam nos critérios de inclusão a que corresponderam na fase de seleção. Ao término desse processo, 8 artigos foram rejeitados, restando apenas 16 aceitos. Com a leitura dos trabalhos, foi possível encontrar as respostas para as perguntas de pesquisa definidas no protocolo. As arquiteturas identificadas no desenvolvimento das ferramentas foram a de filas de trabalho, como proposto por [H et al. 2013] e a de agendador de trabalhos, proposto por [Wang et al. 2013].

Foram identificadas várias tecnologias utilizadas no desenvolvimento desse tipo de ferramenta. [Ben Guebila et al. 2021] utilizou Django e Python para o desenvolvimento de sua ferramenta, assim como [van Wyk et al. 2021], que utilizou o framework Flask. Outros autores como [D et al. 2014] utilizaram PHP no desenvolvimento de seus trabalhos. Alguns autores utilizaram scripts em R para calcular os parâmetros de formação das redes regulatórias de genes, como [H et al. 2013], [Aj et al. 2017], [Wang et al. 2013] e [Ding et al. 2021]. Alguns dos bancos de dados utilizados foram o SQLite, como podemos no trabalho proposto por [Ben Guebila et al. 2021], MySQL em [Wang et al. 2013] e o Redis, visto em [van Wyk et al. 2021].

O Cytoscape foi o framework mais utilizado para a visualização das redes regulatórias de genes formadas a partir do processamento dos dados. Como no artigo [noa 2019] o utilizaram para mostrar seus resultados. O Vis.js foi utilizado por [Ben Guebila et al. 2021] para o mesmo propósito.

Uma das funcionalidades observadas foi a de point-and-click, como podemos ver em [van Wyk et al. 2021].

Os principais desafios que serão enfrentados na construção da plataforma são questões de desempenho, já que a quantidade de dados para a criação das redes regulatórias de genes pode ser muito grande e conseqüentemente exigirá um consumo alto de recursos computacionais e conjuntamente a experiência de usuário pois, o sistema deverá ser amigável até para pessoas com pouca experiência em tecnologia.

### 4. Conclusões

A bioinformática é relativamente recente e segue se desenvolvendo, sendo de extrema importância para o desenvolvimento humano, possuindo ainda diversas questões a serem apresentadas e respondidas. A implementação de redes regulatórias de genes vêm demonstrando resultados importantes e contribuindo fortemente para a construção de conhecimento nas ciências da vida.

Após a realização desta revisão sistemática podemos notar uma possível lacuna, presente em diversas aplicações desenvolvidas e apresentadas nos artigos lidos, que é o fato das aplicações limitarem a construção de um número pequeno de redes regulatórias, definindo apenas alguns organismos específicos para a demonstração de resultados, por exemplo, o [van Wyk et al. 2021] que mostrou apenas o *Plasmodium falciparum* causador da malária.

## Referências

- (2019). Development and use of a Cytoscape app for GRNCOP2. *Computer Methods and Programs in Biomedicine*, 177:211–218.
- Aj, M., C, D., S, C.-R., Ds, H., and T, P.-A. (2016). Graphlet Based Metrics for the Comparison of Gene Regulatory Networks. *PloS one*, 11(10).
- Aj, M., S, C.-R., C, D., and T, P.-A. (2017). LoTo: a graphlet based method for the comparison of local topology between gene regulatory networks. *PeerJ*, 5.
- Ben Guebila, M., Lopes-Ramos, C. M., Weighill, D., Sonawane, A., Burkholz, R., Shamsaei, B., Platig, J., Glass, K., Kuijjer, M., and Quackenbush, J. (2021). GRAND: a database of gene regulatory network models across human conditions. *Nucleic Acids Research*, 50(D1):D610–D621.
- D, G., J, S., Z, Z., P, W., J, Q., Y, D., Kr, B., J, W., and B, Y. (2014). PTHGRN: unraveling post-translational hierarchical gene regulatory networks using PPI, ChIP-seq and gene expression data. *Nucleic acids research*, 42(Web Server issue).
- Ding, J., Blencowe, M., Nghiem, T., Ha, S.-M., Chen, Y.-W., Li, G., and Yang, X. (2021). Mergeomics 2.0: a web server for multi-omics data integration to elucidate disease networks and predict therapeutics. *Nucleic Acids Research*, 49(W1):W375–W387.
- H, X., H, Y., K, T., Q, S., C, W., Yy, L., and Yx, L. (2013). cGRNB: a web server for building combinatorial gene regulatory networks through integrated engineering of seed-matching sequence information and gene expression datasets. *BMC systems biology*, 7 Suppl 2(Suppl 2).
- Hollander, M., Hamed, M., Helms, V., and Neininger, K. (2018). MutaNET: a tool for automated analysis of genomic mutations in gene regulatory networks. *Bioinformatics*, 34(5):864–866.
- Kharumnuid, G. and Roy, S. (2015). Tools for in-Silico Reconstruction and Visualization of Gene Regulatory Networks (GRN). In *2015 Second International Conference on Advances in Computing and Communication Engineering*, pages 421–426.
- Kulkarni, S. R., Vanechoutte, D., Van de Velde, J., and Vandepoele, K. (2018). TF2Network: predicting transcription factor regulators and gene regulatory networks in Arabidopsis using publicly available binding site information. *Nucleic Acids Research*, 46(6):e31.
- L, P., Mi, Z., D, V., M, M., I, M., Da, S., Gn, G., and E, E. (2021). Demetra Application: An integrated genotype analysis web server for clinical genomics in endometriosis. *International journal of molecular medicine*, 47(6).
- M, Z., Q, L., D, Y., B, Y., W, G., Y, X., and G, X. (2019). GeNeCK: a web server for gene network construction and visualization. *BMC bioinformatics*, 20(1).
- Ma, A., Wang, C., Chang, Y., Brennan, F. H., McDermaid, A., Liu, B., Zhang, C., Popovich, P. G., and Ma, Q. (2020). IRIS3: integrated cell-type-specific regulon inference server from single-cell RNA-Seq. *Nucleic Acids Research*, 48(W1):W275–W286.

- Mercatelli, D., Scalambra, L., Triboli, L., Ray, F., and Giorgi, F. M. (2020). Gene regulatory network inference resources: A practical overview. *Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms*, 1863(6):194430.
- N, S., Y, T., G, Y., and Y, O. (2022). CBNplot: Bayesian network plots for enrichment analysis. *Bioinformatics (Oxford, England)*.
- Neal, M. L., Wei, L., Peterson, E., Arrieta-Ortiz, M. L., Danziger, S., Baliga, N., Kaushansky, A., and Aitchison, J. (2021). A systems-level gene regulatory network model for *Plasmodium falciparum*. *Nucleic Acids Research*, 49(9):4891–4906.
- Nguyen, H., Tran, D., Tran, B., Pehlivan, B., and Nguyen, T. (2020). A comprehensive survey of regulatory network inference methods using single cell RNA sequencing data. *Briefings in Bioinformatics*, 22(3):bbaa190.
- R, S., M, S., M, G., R, S., and L, V. d. B. (2021). Spatiotemporal Gene Expression Profiling and Network Inference: A Roadmap for Analysis, Visualization, and Key Gene Identification. *Methods in molecular biology (Clifton, N.J.)*, 2328.
- Sa, K., B, A., Cx, W., and A, C. (2015). FlyOde - a platform for community curation and interactive visualization of dynamic gene regulatory networks in *Drosophila* eye development. *F1000Research*, 4.
- van Wyk, R., van Biljon, R., and Birkholtz, L.-M. (2021). MALBoost: a web-based application for gene regulatory network analysis in *Plasmodium falciparum*. *Malaria Journal*, 20(1):317.
- W, C., W, Z., A, Y., A, X., H, W., M, C., T, L., P, W., and H, Y. (2017). Integrated analysis of microRNA and gene expression profiles reveals a functional regulatory module associated with liver fibrosis. *Gene*, 636.
- Wang, M., Verdier, J., Benedito, V. A., Tang, Y., Murray, J. D., Ge, Y., Becker, J. D., Carvalho, H., Rogers, C., Udvardi, M., and He, J. (2013). LegumeGRN: A Gene Regulatory Network Prediction Server for Functional and Comparative Studies. *PLoS ONE*, 8(7):e67434.
- Weinandt, N., Jackson, L. M., Gnimpieba, E. Z., and Lushbough, C. (2014). Pheno2GRN: a workflow for phenotype to gene network study and reverse engineering comparison. In *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics, BCB '14*, pages 797–804, New York, NY, USA. Association for Computing Machinery.
- X, C., L, H., Y, L., Y, L., W, S., B, S., W, Z., Y, Z., and H, W. (2021). siGCD: a web server to explore survival interaction of genes, cells and drugs in human cancers. *Briefings in bioinformatics*, 22(5).

## 5. Planilha-resumo de Resultados

Abaixo segue a planilha resumo com os artigos selecionados e excluídos, juntamente com suas respectivas avaliações e descrições.

**Tabela 2. Planilha-resumo dos trabalhos selecionados.**

Identificação do Trabalho	I1	I2	E1	E2	E3	E4	E5	Descrição	Avaliação
Mercatelli, D., Scalambra, L., Triboli, L., Ray, F., and Giorgi, F. M. (2020). Gene regulatory network inference resources: A practical overview. <i>Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms</i> , 1863(6):194430	X							O artigo realiza um estudo amplo das principais ferramentas e métodos utilizados para a visualização e construção das redes de genes reguladores.	O artigo será útil para entender melhor as ferramentas existentes e suas funcionalidades.
van Wyk, R., van Biljon, R., and Birkholtz, L.-M. (2021). MALBoost: a web-based application for gene regulatory network analysis in <i>Plasmodium falciparum</i> . <i>Malaria Journal</i> , 20(1):317		X						O artigo descreve a ferramenta MALBoost, criada para analisar a GRN do protozoário <i>Plasmodium falciparum</i> , que é o causador da malária.	O artigo além de falar sobre a construção de um aplicativo utilizando python e o framework flask, mostra também a arquitetura utilizada pelo programa.
(2019). Development and use of a Cytoscape app for GRNCOP2. <i>Computer Methods and Programs in Biomedicine</i> , 177:211–218		X						Este artigo demonstra de maneira resumida a implementação e o ecossistema do GRNCOP2, usando o Cytoscape para visualização de redes regulatórias de genes, utilizando a doença de Alzheimer's como exemplo.	O artigo demonstra alguns aspectos da implementação e cita ainda uma ferramenta que facilita a visualização de dados das redes regulatórias.

Continua na próxima página

**Tabela 2 – continuação da página anterior**

<b>Identificação do Trabalho</b>	<b>I1</b>	<b>I2</b>	<b>E1</b>	<b>E2</b>	<b>E3</b>	<b>E4</b>	<b>E5</b>	<b>Descrição</b>	<b>Avaliação</b>
N, S., Y, T., G, Y., and Y, O. (2022). CBNplot: Bayesian network plots for enrichment analysis. <i>Bioinformatics (Oxford, England)</i>						X		O artigo apresenta uma ferramenta chamada CBNplot, desenvolvida em R, que faz inferência da rede Bayesiana de dados de expressão gênica.	O artigo não será aproveitado por não estar relacionado com a visualização de redes regulatórias de genes.
Aj, M., S, C.-R., C, D., and T, P.-A. (2017). LoTo: a graphlet based method for the comparison of local topology between gene regulatory networks. <i>PeerJ</i> , 5				X			X	O artigo descreve uma metodologia de comparação entre diferentes redes de genes reguladores, chamada de LoTo.	O artigo não realiza nenhuma implementação, e tem como principal foco a demonstração de resultados matemáticos utilizando a LoTo.
Sa, K., B, A., Cx, W., and A, C. (2015). FlyOde - a platform for community curation and interactive visualization of dynamic gene regulatory networks in <i>Drosophila</i> eye development. <i>F1000Research</i> , 4		X						O artigo apresenta a ferramenta FlyOde, utilizada para visualização de redes regulatórias de genes de uma base de dados predefinida. Possui o objetivo de expandir a base de dados disponível, para permitir que sejam feitas comparações de redes regulatórias de genes de organismos diferentes.	O artigo descreve o objetivo de sua ferramenta e tecnologias utilizadas para a visualização das redes, bem como o objetivo de sua ferramenta, o que ajuda a definir quais são os requisitos indispensáveis para esse tipo de ferramenta.

Continua na próxima página

**Tabela 2 – continuação da página anterior**

<b>Identificação do Trabalho</b>	<b>I1</b>	<b>I2</b>	<b>E1</b>	<b>E2</b>	<b>E3</b>	<b>E4</b>	<b>E5</b>	<b>Descrição</b>	<b>Avaliação</b>
Aj, M., C, D., S, C.-R., Ds, H., and T, P.-A. (2016). Graphlet Based Metrics for the Comparison of Gene Regulatory Networks. <i>PLoS one</i> , 11(10)		X						O artigo apresenta a visualização de redes regulatórias de genes através do Cytoscape. Foi utilizado um módulo em R para fazer os cálculos de geração da rede.	O artigo descreve qual ferramenta foi utilizada para a visualização de redes regulatórias de genes.
W, C., W, Z., A, Y., A, X., H, W., M, C., T, L., P, W., and H, Y. (2017). Integrated analysis of microRNA and gene expression profiles reveals a functional regulatory module associated with liver fibrosis. <i>Gene</i> , 636					X			O artigo apresenta um estudo sobre fibrose hepática.	O artigo não será aproveitado por não estar relacionado a redes regulatórias de genes.
H, X., H, Y., K, T., Q, S., C, W., Yy, L., and Yx, L. (2013). cGRNB: a web server for building combinatorial gene regulatory networks through integrated engineering of seed-matching sequence information and gene expression datasets. <i>BMC systems biology</i> , 7 Suppl 2(Suppl 2)		X						O artigo apresenta o cGRNB, um web-server que utiliza a linguagem PHP para realizar a integração com dois módulos escritos em R para o processamento de dados, que resultam em redes regulatórias de genes.	O artigo ajudará a entender quais tecnologias são utilizadas no desenvolvimento desse tipo de ferramenta.

Continua na próxima página

**Tabela 2 – continuação da página anterior**

<b>Identificação do Trabalho</b>	<b>I1</b>	<b>I2</b>	<b>E1</b>	<b>E2</b>	<b>E3</b>	<b>E4</b>	<b>E5</b>	<b>Descrição</b>	<b>Avaliação</b>
D, G., J, S., Z, Z., P, W., J, Q., Y, D., Kr, B., J, W., and B, Y. (2014). PTHGRN: unravelling post-translational hierarchical gene regulatory networks using PPI, ChIP-seq and gene expression data. <i>Nucleic acids research</i> , 42(Web Server issue)		X						O artigo apresenta a ferramenta PTHGRN, um web-server escrito em PHP, que suporta scripts escritos em Pascal. O Cytoscape foi utilizado para visualização das redes regulatórias. O usuário recebe um e-mail com o link para o resultado do trabalho solicitado.	O artigo serve de base para entender as linguagens e arquiteturas que podem ser utilizadas para aplicações que visualizam redes regulatórias de genes.
L, P., Mi, Z., D, V., M, M., I, M., Da, S., Gn, G., and E, E. (2021). Demetra Application: An integrated genotype analysis web server for clinical genomics in endometriosis. <i>International journal of molecular medicine</i> , 47(6)						X		O artigo apresenta uma ferramenta baseada na web para auxiliar médicos especialistas e pesquisadores no processo de diagnóstico de endometriose.	O artigo não está relacionado ao tema e portanto não será apresentado.
X, C., L, H., Y, L., Y, L., W, S., B, S., W, Z., Y, Z., and H, W. (2021). siGCD: a web server to explore survival interaction of genes, cells and drugs in human cancers. <i>Briefings in bioinformatics</i> , 22(5)							X	O artigo demonstra uma aplicação web que tem como objetivo entender a interação, entre drogas e células em alguns tipos de câncer.	Este artigo não será aproveitado pois não apresenta a criação de nenhuma interface para visualização das redes regulatórias de genes.

Continua na próxima página

**Tabela 2 – continuação da página anterior**

<b>Identificação do Trabalho</b>	<b>I1</b>	<b>I2</b>	<b>E1</b>	<b>E2</b>	<b>E3</b>	<b>E4</b>	<b>E5</b>	<b>Descrição</b>	<b>Avaliação</b>
R, S., M, S., M, G., R, S., and L, V. d. B. (2021). Spatiotemporal Gene Expression Profiling and Network Inference: A Roadmap for Analysis, Visualization, and Key Gene Identification. <i>Methods in molecular biology (Clifton, N.J.)</i> , 2328				X				O artigo apresenta um guia para análise, visualização e identificação de genes chave.	O artigo completo não foi encontrado para leitura. Através do resumo é possível perceber que o conteúdo não se enquadra no tema proposto para a revisão.
M, Z., Q, L., D, Y., B, Y., W, G., Y, X., and G, X. (2019). GeNeCK: a web server for gene network construction and visualization. <i>BMC bioinformatics</i> , 20(1)						X		O artigo descreve sobre a ferramenta GeNeCK, que é um web server que permite ao usuário escolher diferentes métodos para a construção de uma GNR;	Apesar do artigo mostrar a ferramenta e um pouco da sua interface, seu foco é nos métodos matemáticos utilizados para a construção das redes.
Ben Guebila, M., Lopes-Ramos, C. M., Weighill, D., Sonawane, A., Burkholz, R., Shamsaei, B., Platig, J., Glass, K., Kuijjer, M., and Quackenbush, J. (2021). GRAND: a database of gene regulatory network models across human conditions. <i>Nucleic Acids Research</i> , 50(D1):D610–D621		X						O artigo apresenta GRAND, um banco de dados de redes regulatórias de genes selecionadas por curadores, com mais de 6TB de dados hospedados em nuvem. Desenvolvido com Python e Django, SQLite, Vis.js para visualização das redes regulatórias de genes e hospedado na AWS.	O artigo dá uma visão geral da construção de uma ferramenta para visualização de redes regulatórias de genes.

Continua na próxima página

**Tabela 2 – continuação da página anterior**

<b>Identificação do Trabalho</b>	<b>I1</b>	<b>I2</b>	<b>E1</b>	<b>E2</b>	<b>E3</b>	<b>E4</b>	<b>E5</b>	<b>Descrição</b>	<b>Avaliação</b>
Nguyen, H., Tran, D., Tran, B., Pehlivan, B., and Nguyen, T. (2020). A comprehensive survey of regulatory network inference methods using single cell RNA sequencing data. <i>Briefings in Bioinformatics</i> , 22(3):bbaa190				X			X	O artigo realiza um estudo comparativo entre os principais métodos para a geração das redes de genes reguladores, analisando principalmente o aspecto de performance.	O artigo é um estudo comparativo de métodos para gerar as grn, não demonstra nenhuma implementação ou visualização das redes reguladores de genes.
Neal, M. L., Wei, L., Peterson, E., Arrieta-Ortiz, M. L., Danziger, S., Baliga, N., Kaushansky, A., and Aitchison, J. (2021). A systems-level gene regulatory network model for <i>Plasmodium falciparum</i> . <i>Nucleic Acids Research</i> , 49(9):4891–4906							X	O artigo apresenta uma abordagem utilizando machine learning para geração de redes regulatórias de genes. O método apresentado utiliza um pacote em R.	O artigo não apresentou nenhuma interface para visualização de redes regulatórias de genes. O artigo não será aproveitado.
Ma, A., Wang, C., Chang, Y., Brennan, F. H., McDermaid, A., Liu, B., Zhang, C., Popovich, P. G., and Ma, Q. (2020). IRIS3: integrated cell-type-specific regulon inference server from single-cell RNA-Seq. <i>Nucleic Acids Research</i> , 48(W1):W275–W286						X		O artigo explica sobre a IRIS3, uma ferramenta web para realizar inferências entre genes de humanos e de ratos, demonstrando visualização de resultados e a infraestrutura da ferramenta.	O artigo mostra algumas questões de implementação porém a visualização de redes reguladoras de genes não é abordada de maneira satisfatória.

Continua na próxima página

**Tabela 2 – continuação da página anterior**

<b>Identificação do Trabalho</b>	<b>I1</b>	<b>I2</b>	<b>E1</b>	<b>E2</b>	<b>E3</b>	<b>E4</b>	<b>E5</b>	<b>Descrição</b>	<b>Avaliação</b>
Hollander, M., Hamed, M., Helms, V., and Neisinger, K. (2018). MutaNET: a tool for automated analysis of genomic mutations in gene regulatory networks. <i>Bioinformatics</i> , 34(5):864–866		X						O artigo descreve de maneira bem resumida, a criação do MutaNET, que é uma ferramenta implementada em python e que tem como objetivo analisar o impacto de mutações na regulação de genes.	O artigo mostra uma sequência resumida de passos, desde a obtenção de dados para análise até os resultados da análise.
Kulkarni, S. R., Vanechoutte, D., Van de Velde, J., and Vandepoele, K. (2018). TF2Network: predicting transcription factor regulators and gene regulatory networks in Arabidopsis using publicly available binding site information. <i>Nucleic Acids Research</i> , 46(6):e31				X				Este artigo demonstra a ferramenta TF2Network, usada para identificar a rede de genes reguladores de uma espécie de planta chamada Arabidopsis.	O artigo apesar de falar de redes reguladoras de genes e demonstrar uma interface de visualização, não possui um foco na implementação da ferramenta e por isso, não responde de maneira satisfatória as questões de pesquisa.

Continua na próxima página

**Tabela 2 – continuação da página anterior**

<b>Identificação do Trabalho</b>	<b>I1</b>	<b>I2</b>	<b>E1</b>	<b>E2</b>	<b>E3</b>	<b>E4</b>	<b>E5</b>	<b>Descrição</b>	<b>Avaliação</b>
Wang, M., Verdier, J., Benedito, V. A., Tang, Y., Murray, J. D., Ge, Y., Becker, J. D., Carvalho, H., Rogers, C., Udvardi, M., and He, J. (2013). LegumeGRN: A Gene Regulatory Network Prediction Server for Functional and Comparative Studies. <i>PLoS ONE</i> , 8(7):e67434		X						O artigo apresenta a ferramenta LegumeGRN, desenvolvida em J2EE, Tomcat, JSP e bibliotecas Javascript para interatividade no front-end. Foram utilizadas requisições AJAX para receber os dados na visualização das redes. Foram utilizados clusters e um job scheduler para responder as requisições dos usuários. Os cálculos para geração das redes foram feitos em módulos escritos em Matlab e R. Cytoscape foi utilizado para gerar a visualização das redes no site. A ferramenta permite customização dos dados de entrada.	O artigo apresenta as tecnologias utilizadas para a criação de uma ferramenta web para visualização de redes regulatórias de genes.
Kharumnuid, G. and Roy, S. (2015). Tools for in-Silico Reconstruction and Visualization of Gene Regulatory Networks (GRN). In <i>2015 Second International Conference on Advances in Computing and Communication Engineering</i> , pages 421–426		X						O artigo apresenta uma comparação de ferramentas escritas em diversas linguagens de programação para visualização de redes regulatórias de genes.	O artigo será útil para entender quais linguagens são mais apropriadas para o desenvolvimento de ferramentas para visualização de redes regulatórias de genes.

Continua na próxima página

**Tabela 2 – continuação da página anterior**

<b>Identificação do Trabalho</b>	<b>I1</b>	<b>I2</b>	<b>E1</b>	<b>E2</b>	<b>E3</b>	<b>E4</b>	<b>E5</b>	<b>Descrição</b>	<b>Avaliação</b>
Weinandt, N., Jackson, L. M., Gnimpieba, E. Z., and Lushbough, C. (2014). Pheno2GRN: a workflow for phenotype to gene network study and reverse engineering comparison. In <i>Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics, BCB '14</i> , pages 797–804, New York, NY, USA. Association for Computing Machinery		X						O artigo demonstra um fluxo de trabalho, desde a obtenção dos dados a serem analisados até a sua visualização. Descrevendo de maneira objetiva os métodos e algumas ferramentas usadas no estudo.	O artigo tem uma seção só falando do framework javascript Cytoscape, utilizado para visualização de redes de genes.
Ding, J., Blencowe, M., Nghiem, T., Ha, S.-M., Chen, Y.-W., Li, G., and Yang, X. (2021). Mergeomics 2.0: a web server for multi-omics data integration to elucidate disease networks and predict therapeutics. <i>Nucleic Acids Research</i> , 49(W1):W375–W387	X							Mergeomics 2.0 é um web-server baseado no pacote Mergeomics, escrito em R. O artigo demonstra a preocupação dos autores em facilitar o uso da ferramenta, citando o aprimoramento da interface de usuário e dos tutoriais disponíveis para utilização e entendimento dos resultados.	O artigo é útil para ajudar a entender o que é preciso para tornar as ferramentas de visualização de redes regulatórias mais amigáveis ao usuário.

**ANEXOS**

## ANEXO A – Documento de Definição de Requisitos

	<p align="center"><b>FunRegulation</b>  <b>Construção e Manipulação de</b>  <b>Redes Regulatórias de Genes</b>  <b>de Fungos</b></p>	
	<p align="center"><b>Documento de Definição de</b>  <b>Requisitos</b></p>	

## Sumário

1.	Introdução	1
2.	Contextualização	1
2.4.1	Principais Funcionalidades do Sistema	3
2.5.1	Interfaces de Hardware	3
2.5.2	Interfaces de Comunicação	3
2.5.3	Interfaces de Sistema	3
2.5.4	Interfaces com Serviços	4
3.	Fornecedores de Requisitos	5
4.	Requisitos do Sistema	6
4.1	Requisitos Funcionais	6
4.2	Requisitos Não-Funcionais	13
4.3	Regras de Negócio	15
5.	Verificação de Requisitos	18
6.	Validação de Requisitos	18

## **1. Introdução**

Este documento tem por objetivo descrever a visão geral do sistema FunRegulation, que compreende uma plataforma para construção e manipulação de redes regulatórias de genes de fungos. Essa descrição inclui a contextualização do ambiente atual, propósito do sistema, descrição de escopo, características negociais e ambientais e solução adotada com seus requisitos funcionais, requisitos não funcionais e regras de negócio.

## **2. Contextualização**

Esta seção descreve a contextualização do ambiente atual, bem como uma visão geral do escopo do projeto FunRegulation.

### **2.1 Processo Atual**

A análise de genomas é uma tarefa complexa e exige grande esforço dos cientistas, principalmente devido ao tamanho dos genomas e à dificuldade de encontrar informações relevantes rapidamente. Dentro desse contexto, as tecnologias da bioinformática surgem como uma forma de processar os dados biológicos de forma rápida e precisa. Apesar da evolução notória da bioinformática nos últimos anos, ainda não existe um sistema web gratuito para construção de redes regulatórias de genes de fungos.

### **2.2 Tecnologias e Sistemas em Uso**

Atualmente não existe um sistema web que dê o suporte necessário para construção e manipulação de redes regulatórias de genes de um fungo, porém existem plataformas, softwares e bibliotecas que compartilham algumas das funcionalidades e que são utilizadas como fonte de requisitos funcionais para este sistema.

- Cytoscape: fornece ferramentas para a visualização e análise de redes complexas. Há também o Cytoscape.js, uma implementação em forma de biblioteca JavaScript para uso no navegador.
- TF2Network: constrói redes de regulação de genes somente para o organismo *Arabidopsis thaliana* através de uma plataforma web. A construção das redes utiliza informações sobre predição de sítios de ligação de fatores de transcrição, interação proteína-proteína (PPI) e informações sobre expressão diferencial de genes. A plataforma permite a visualização e manipulação da rede através do browser.
- OtthoVenn2: plataforma web que permite a escolha de dois ou mais organismos para análise de agrupamentos de genes ortólogos entre as espécies escolhidas. O workflow desse sistema compartilha diversas similaridades com o workflow pretendido para o FunRegulation.
- Ensembl Fungi: Banco de dados gratuito que oferece genomas completos e anotações para diversas espécies de fungos. Essas informações genômicas podem ser importadas para o FunRegulation.

### 2.3 Propósito do Sistema

Atualmente o número de genomas sequenciados vem aumentando consideravelmente. Consequentemente, cresce a demanda para facilitar e agilizar a interpretação desses genomas. A bioinformática e a biologia computacional desempenham um papel chave nesse contexto, permitindo a organização e análise desses dados biológicos a partir da construção de novas ferramentas computacionais.

FunRegulation utilizará uma abordagem computacional com o objetivo de identificar quais os genes do genoma são os reguladores e quais são os genes regulados e posteriormente construir uma Rede de Regulação Gênica (GRN, do inglês Gene Regulatory Network). Uma GRN compreende uma coleção de reguladores moleculares (TFs, do inglês Transcription Factors) que interagem com os genes

regulados (TG, do inglês Target Genes) para regular os níveis de expressão de mRNA e proteínas. O entendimento dessas interações regulatórias é fundamental para compreender processos biológicos intracelulares, fenótipos, doenças, etc.

## 2.4 Visão Geral do Escopo

Esse projeto visa o desenvolvimento de um sistema para a construção e manipulação de GRNs de um fungo.

### 2.4.1 Principais Funcionalidades do Sistema

- Manter usuários;
- Construir uma GRN;
- Manipular e configurar uma GRN;
- Manter *Views* de uma GRN;
- Exportar uma GRN.

## 2.5 Relacionamentos Ambientais

### 2.5.1 Interfaces de Hardware

O modelo cliente-servidor será composto por um servidor Unix e o hardware dos clientes poderá ser qualquer computador Desktop, Notebook ou aparelho móvel que possua um sistema operacional com navegador.

### 2.5.2 Interfaces de Comunicação

A comunicação entre clientes e servidor ocorrerá via internet através do protocolo HTTP a partir de um navegador.

### 2.5.3 Interfaces de Sistema

O software será um sistema web, modelo cliente-servidor, ou seja, o acesso poderá ocorrer a partir de qualquer sistema operacional que tenha um navegador e esteja conectado à internet.

O servidor deverá comportar um servidor web, um servidor de banco de dados e a instalação do software ProteinOrtho, que será utilizado para mapeamento de genes ortólogos.

#### 2.5.4 Interfaces com Serviços

O sistema utilizará o serviço web RSAT com comunicação via SOAP para a predição de sítios de ligação de fatores de transcrição:

[http://rsat.sb-roscoff.fr/web\\_services/RSATWS.wsdl](http://rsat.sb-roscoff.fr/web_services/RSATWS.wsdl)

#### 2.6 Características dos Usuários

Em sua grande maioria, os usuários que utilizarão o FunRegulation serão pesquisadores, professores e alunos de nível superior. Consequentemente, possuem experiência com biologia e possuem conhecimentos técnicos para utilizar a aplicação.

#### 2.7 Limitações do Sistema

- A plataforma não terá suporte online.

#### 2.8 Premissas

- A aplicação será desenvolvida na linguagem Python;
- A aplicação será desenvolvida para a web;
- A interface da aplicação será em inglês.

#### 2.9 Definições, Acrônimos e Abreviações

<b>Sigla</b>	<b>Definição</b>
<b>GRN</b>	Rede de Regulação Gênica (Gene Regulatory Network)
<b>Genoma</b>	Sequência completa de DNA (ácido desoxirribonucleico) de um organismo, ou seja, o conjunto de todos os genes de um organismo.
<b>Gene annotation</b>	Identificação da localização e atribuição

	de funções para cada gene de um genoma.
<b>Symbol gene</b>	Abreviação do nome de um determinado gene.
<b>Locus tag</b>	Identificadores aplicados sistematicamente a cada gene em um projeto de sequenciamento.
<b>TF</b>	Fator de Transcrição (Transcription Factor) - proteínas que ajudam a regular a transcrição de TGs através da conexão ao DNA próximo ao TG.
<b>TG</b>	Gene alvo (Target Gene). - São os genes regulados pelos TF.

### 3. Fornecedores de Requisitos

**Nome: Alexandre Rafael Lenz**

Função no projeto: Orientador

e-mail: alenz@uneb.br

telefone: (71) 99911-9433

**Nome: Kevin Oliveira da Silva**

Função no projeto: Desenvolvedor do sistema

e-mail: kevin.oliveira@live.com

telefones: (71) 99199-6405

#### 4. Requisitos do Sistema

Esta seção detalha todos os requisitos funcionais, requisitos não funcionais e regras de negócio identificados durante a fase de especificação do sistema.

##### 4.1 Requisitos Funcionais

Para cada especificação de requisito funcional, são indicadas a sua origem e descrição.

RF01 - Cadastrar usuário		
<b>Descrição:</b>	O sistema deve permitir o cadastro de usuários a partir dos seguintes campos: <ul style="list-style-type: none"><li>• Nome completo;</li><li>• E-mail institucional;</li><li>• Senha;</li><li>• Nome da instituição;</li><li>• País;</li><li>• Se for um usuário brasileiro, selecionar o estado.</li></ul>	
<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>	Leandro Lopes	
<b>Versão:</b> 2.0	<b>Status:</b> Aprovado	<b>Prioridade:</b> Média
<b>RNs</b>	RN02; RN03;	

RF02 - Confirmar cadastro	
<b>Descrição:</b>	O sistema deve enviar um e-mail para confirmação do cadastro do usuário. No corpo do e-mail deve constar um link redirecionando o usuário para uma tela no sistema, onde o cadastro será confirmado e o usuário terá acesso às funcionalidades de um usuário com login.

<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>	Leandro Lopes	
<b>Versão: 2.0</b>	<b>Status: Aprovado</b>	<b>Prioridade: Média</b>

<b>RF03 – Efetuar Log in</b>		
<b>Descrição:</b>	O usuário deve informar seu e-mail e senha para efetuar o log in no sistema. Após o login o usuário terá acesso aos RF06 a RF18.	
<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>	Leandro Lopes	
<b>Versão: 2.0</b>	<b>Status: Aprovado</b>	<b>Prioridade: Média</b>

<b>RF04 – Efetuar Log out</b>		
<b>Descrição:</b>	O usuário pode realizar log out para finalizar seu acesso ao sistema.	
<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>	Leandro Lopes	
<b>Versão: 2.0</b>	<b>Status: Aprovado</b>	<b>Prioridade: Média</b>
<b>RNs</b>	RN09	

<b>RF05 - Recuperar senha</b>		
-------------------------------	--	--

<b>Descrição:</b>	Caso o usuário perca sua senha e deseje recuperá-la, o sistema deverá enviar um e-mail com um link para a tela de redefinição da senha. Nesta tela o usuário deverá preencher os campos Nova senha e Repetir nova senha. Após confirmar, a senha será redefinida.	
<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>	Leandro Lopes	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Média
<b>RNs</b>	RN01	

<b>RF06 - Construir GRN</b>	
<b>Descrição:</b>	<p>O sistema deve permitir a construção de uma GRN conforme o fluxo a seguir:</p> <ol style="list-style-type: none"> <li>1. Escolher organismo conforme RF07 ou RF08</li> <li>2. Selecionar interações regulatórias que devem ser consideradas: <ol style="list-style-type: none"> <li>a. Quando a opção “Somente ortologia de organismo modelo” for selecionada, o sistema deve: <ol style="list-style-type: none"> <li>i. Identificar os genes ortólogos utilizando o software ProteinOrtho;</li> <li>ii. Buscar as relações comprovadas entre TFs e TGs no organismo modelo;</li> <li>iii. Quando são encontrados os ortólogos para TF e TG de uma interação, uma nova interação TF-TG é criada para o organismo alvo.</li> </ol> </li> <li>b. Quando a opção “Ortologia de organismo modelo e sítios de ligação dos fatores de transcrição” for selecionada, o sistema deve: <ol style="list-style-type: none"> <li>i. Executar itens i a iii descritos acima;</li> <li>ii. Extrair os promotores 1000bp upstream de cada TG do genoma do organismo alvo;</li> <li>iii. Obter as matrizes PWM do banco de dados CIS-BP;</li> <li>iv. Realizar a predição dos sítios de ligação dos fatores de transcrição a partir do web service RSAT;</li> </ol> </li> </ol> </li> </ol>

	<p>v. Compilar os resultados do RSAT;</p> <p>3. Construir a rede de interações regulatórias de genes.</p> <p>O sistema deverá mostrar os passos que estão sendo executados para que o usuário acompanhe a construção da GRN. Quando a construção da GRN for concluída o sistema aciona o RF09 para exibir a GRN.</p>	
<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>	Leandro Lopes	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Alta

RF07 – Carregar informações genômicas de Ensembl Fungi		
<b>Descrição:</b>	<p>Para construir uma GRN, o sistema deverá prover uma lista de organismos obtidos da plataforma Ensembl Fungi. O sistema deve permitir a seleção de um organismo para construção da GRN e para o organismo escolhido, o sistema deve obter da plataforma Ensembl Fungi:</p> <ul style="list-style-type: none"> <li>● Genoma completo no formato fasta</li> <li>● Proteínas no formato fasta</li> <li>● Anotação genômica no formato gff3</li> <li>● Anotação funcional das proteínas contendo GO terms e/ou InterPro</li> </ul>	
<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>	Leandro Lopes	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Média
<b>RNs</b>	RN01; RN04; RN05; RN08	

RF08 - Carregar informações genômicas de arquivos locais		
<b>Descrição:</b>	Para construir uma GRN, caso o usuário não encontre o organismo desejado na lista obtida da plataforma Ensembl Fungi, o sistema deverá permitir o upload dos arquivos de informações genômicas: <ul style="list-style-type: none"> <li>● Genoma completo no formato fasta</li> <li>● Proteínas no formato fasta</li> <li>● Anotação genômica no formato gff3</li> <li>● Anotação funcional no formato tsv (deve conter InterPro e/ou GO terms)</li> </ul>	
<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>	Leandro Lopes	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Alta
<b>Rastreamento</b>	RN01; RN05.	

RF09 – Exibir e Manipular GRN		
<b>Descrição:</b>	O sistema deve exibir a GRN construída permitindo a visualização gráfica (RN06 e RN07) e manipulação de suas interações regulatórias. <ul style="list-style-type: none"> <li>● O usuário poderá selecionar um nó da GRN para ver sua anotação funcional, conforme RF10;</li> <li>● O usuário poderá exibir estatísticas gerais da GRN, conforme RF11;</li> </ul>	
<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>		
<b>Versão: 1.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Alta
<b>RNs</b>	RN06; RN07;	

RF10 – Exibir anotação funcional de um nó		
<b>Descrição:</b>	O sistema deve exibir informações de anotação funcional quando o usuário selecionar um nó (TF ou TG).	
<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>	Jéssica Cardoso	
<b>Versão:</b> 2.0	<b>Status:</b> Aprovado	<b>Prioridade:</b> Média

RF11 - Exibir estatísticas		
<b>Descrição:</b>	O sistema deve exibir um modal com as estatísticas gerais da GRN ou de uma <i>View</i> .	
<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>	Jéssica Cardoso	
<b>Versão:</b> 2.0	<b>Status:</b> Aprovado	<b>Prioridade:</b> Média

RF12 – Configurar GRN	
<b>Descrição:</b>	<p>O sistema deve permitir que o usuário configure a GRN conforme suas expectativas, de acordo com a RN08:</p> <ul style="list-style-type: none"> <li>● O usuário poderá escolher quais genes serão exibidos na GRN a partir de uma lista de TFs com o symbol gene e locus tag, podendo selecionar um ou mais TFs para exibição na GRN;</li> <li>● O usuário poderá ampliar ou reduzir o tamanho da GRN, conforme RF13;</li> <li>● O usuário poderá escolher o layout de exibição da GRN, conforme RF14;</li> <li>● O usuário poderá filtrar quais interações regulatórias devem ser mostradas na GRN, conforme RF15;</li> </ul>

<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>		
<b>Versão: 1.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Baixa
<b>RNs</b>	RN08;	

<b>RF13 - Permitir zoom</b>		
<b>Descrição:</b>	<p>O sistema deve permitir a ampliação ou redução da GRN a partir de uma funcionalidade de zoom. A funcionalidade deve ser acessível por:</p> <ul style="list-style-type: none"> <li>● Scroll do mouse;</li> <li>● Pinça no touchpad;</li> <li>● teclas de atalho + e -;</li> <li>● botões +, - e fit-to-window.</li> </ul>	
<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>	Jéssica Cardoso	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Baixa

<b>RF14 - Escolher Layout de Exibição</b>		
<b>Descrição:</b>	As redes devem ter layouts de exibição predefinidos e o usuário pode mudar os estilos da rede.	
<b>Origem:</b>	Alexandre Lenz e Kevin Lima	
<b>Modificado por:</b>		
<b>Versão: 1.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Baixa

--	--	--

RF15 - Filtrar interações regulatórias		
<b>Descrição:</b>	O sistema deve permitir que o usuário filtre as arestas que serão mostradas na GRN, conforme RN07.	
<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>	Leandro Lopes	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Muito baixa
<b>RNs</b>	RN07	

RF16 – Manter GRNs e Views		
<b>Descrição:</b>	<p>O sistema deve mostrar a lista de GRNs que um usuário criou. O sistema também deve permitir que o usuário renomeie e exclua GRNs. No caso da exclusão de uma GRN, todas as Views associadas também serão excluídas.</p> <p>Cada GRN deve ter sua lista de Views, de maneira hierárquica como se fossem filhas.</p> <p>Uma View corresponde às configurações efetuadas (RF12 a RF15) para uma GRN. A GRN original nunca é alterada.</p> <p>O sistema deve permitir que o usuário crie, edite (renomear e modificar configurações), visualize e exclua Views.</p>	
<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>	Jéssica Cardoso	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Alta

RF17 - Exportar GRN ou <i>View</i>		
<b>Descrição:</b>	O sistema deve permitir a exportação de uma GRN ou de uma <i>View</i> nos formatos tsv, pdf, png e svg.	
<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>	Jéssica Cardoso	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Alta

RF18 - Mostrar tutorial		
<b>Descrição:</b>	O sistema deve exibir um tutorial de uso da plataforma contendo: <ul style="list-style-type: none"> <li>● Exemplos de arquivos de entrada; <ul style="list-style-type: none"> <li>○ arquivos tsv, fasta e gff3;</li> </ul> </li> <li>● Workflow de execução;</li> <li>● Resultados esperados;</li> <li>● Como analisar resultados.</li> </ul>	
<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>	Leandro Lopes	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Baixa

RF19 - Mostrar tela inicial
-----------------------------

<b>Descrição:</b>	O sistema deverá exibir uma página inicial detalhando o propósito e o workflow.  O sistema também deve mostrar informações de contato: <ul style="list-style-type: none"> <li>● E-mail para problemas científicos;</li> <li>● E-mail para problemas técnicos;</li> <li>● Citação do artigo publicado.</li> </ul>	
<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>		
<b>Versão: 1.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Baixa

#### 4.2 Requisitos Não-Funcionais

Para cada especificação de requisito não-funcional, são indicadas a sua origem, tipo, descrição e critérios de aceitação.

<b>RNF01 – Atualização da lista de organismos</b>		
<b>Descrição:</b>	A cada seis meses deverá ser realizada uma verificação na plataforma Ensembl Fungi para atualizar a lista de organismos disponíveis no sistema.	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Média
<b>Origem:</b>	Alexandre Rafael Lenz	
<b>Modificado por:</b>	Jéssica Cardoso	
<b>Tipo:</b>	Interface Externa	

<b>Cr�terios de Aceita�o:</b>	Todos os organismos da plataforma Ensembl Fungi devem estar dispon�veis no sistema, respeitando RN-4.
-------------------------------	---

<b>RNF02 - Usu�rios Simult�neos</b>		
<b>Descri�o:</b>	O sistema dever� suportar processamento multiusu�rio, ou seja, suportar no m�ximo 10 usu�rios conectados e operando o sistema simultaneamente.	
<b>Vers�o: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> M�dia
<b>Origem:</b>	J�ssica Cardoso	
<b>Modificado por:</b>	J�ssica Cardoso	
<b>Tipo:</b>	Performance	
<b>Cr�terios de Aceita�o:</b>	Teste de Acesso: O sistema deve permitir 10 usu�rios simult�neos, sendo que acima dessa quantidade os usu�rios que tentarem efetuar o login ser�o alertados que devem esperar at� que a carga do sistema diminua.	

<b>RNF03 - Tempo de Resposta</b>		
<b>Descri�o:</b>	Cada consulta ou intera�o com o sistema n�o deve ultrapassar dez segundos.	
<b>Vers�o: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> M�dia

<b>Origem:</b>	Jéssica Cardoso
<b>Modificado por:</b>	Jéssica Cardoso
<b>Tipo:</b>	Performance
<b>Critérios de Aceitação:</b>	Teste de performance: As consultas deverão ser processadas no limite de dez segundos pelo sistema.

RNF04 - Log auditável		
<b>Descrição:</b>	As principais operações realizadas por usuários deverão ser armazenadas em um Log: RF01; RF02; RF03; RF05; RF06; RF16 e RF17. Erros de execução e exceções deverão ser enviadas via e-mail para o responsável técnico do sistema.	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Média
<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>	Alexandre Lenz	
<b>Tipo:</b>	Manutenibilidade	
<b>Critérios de Aceitação:</b>	Teste de performance: As consultas deverão ser processadas no limite de dez segundos pelo sistema.	

#### 4.3 Regras de Negócio

Para cada especificação de regra de negócio, são indicadas a sua origem e descrição.

RN01 - Acessar sistema		
<b>Descrição:</b>	RFs 06 a 17 somente poderão ser executados por usuários logados no sistema.	
<b>Versão: 1.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Média
<b>Origem:</b>	Alexandre Lenz e Kevin Lima	
<b>Modificado por:</b>		






RN02 - Efetuar cadastro		
<b>Descrição:</b>	O cadastro só será efetuado se a senha inserida for de no mínimo 6 dígitos.	
<b>Versão: 1.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Média
<b>Origem:</b>	Alexandre Lenz e Kevin Lima	
<b>Modificado por:</b>		

RN03 - Validar e-mail		
<b>Descrição:</b>	Validar se o e-mail é institucional	
<b>Versão: 1.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Baixa
<b>Origem:</b>	Alexandre Lenz e Kevin Lima	
<b>Modificado por:</b>		

RN04 - Importar organismos		
<b>Descrição:</b>	Os organismos importados da plataforma Ensembl Fungi contemplam somente as 4 ordens: <i>Eurotiales</i> , <i>Sordarialises</i> , <i>Saccharomycetales</i> e <i>Hypocreales</i> .	
<b>Versão: 1.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Média
<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>		

RN05 - Upload de arquivos		
<b>Descrição:</b>	<p>Só será permitido o upload de arquivos de um organismo que faça parte das 4 ordens listadas na RN04.</p> <p>O usuário deverá fazer o upload obrigatório de 4 arquivos:</p> <ol style="list-style-type: none"> <li>1. Genoma completo no formato fasta;</li> <li>2. Anotação de genes no formato gff3;</li> <li>3. Proteínas no formato fasta;</li> <li>4. Anotação funcional no formato TSV <ol style="list-style-type: none"> <li>a. Colunas obrigatórias: GO terms ou InterPro</li> </ol> </li> </ol>	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Alta
<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>	Alexandre Lenz	

RN06 - Identificar elementos da GRN		
-------------------------------------	--	--

<b>Descrição:</b>	<p>Na exibição da GRN, o elemento gráfico que representa cada nó (gene) deve conter o <b>symbol gene</b> (obtido do arquivo de anotação funcional) ou, caso não seja encontrado o symbol gene, deve conter o <b>locus tag</b> (obtido do arquivo de proteínas).</p> <p>Os elementos gráficos devem ser representados conforme segue:</p> <ol style="list-style-type: none"> <li>1) Nós reguladores ou fatores de transcrição (TFs) são representados como losangos vermelhos: </li> <li>2) Nós regulados ou genes alvo (TGs) são representados como elementos ovais azuis: </li> <li>3) Interações (arestas) positivas são: </li> <li>4) Interações (arestas) negativas são: </li> <li>5) Interações (arestas) <i>unknown</i> são: </li> </ol>	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Alta
<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>	Alexandre Lenz	

<b>RN07 - Exibir arestas em cores distintas</b>		
<b>Descrição:</b>	<p>As interações (arestas) devem ser mostradas em cores distintas:</p> <ol style="list-style-type: none"> <li>1) Verde - Somente Ortologia;</li> <li>2) Preto - Protein-DNA* e Ortologia;</li> </ol> <p>* Protein-DNA corresponde aos sítios de ligação de fatores de transcrição preditos;</p>	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Alta
<b>Origem:</b>	Alexandre Lenz	

<b>Modificado por:</b>	Alexandre Lenz
------------------------	----------------

<b>RN08 – Salvar configurações automaticamente</b>		
<b>Descrição:</b>	As configurações realizadas em uma GRN devem ser salvas como uma View automaticamente após 30 segundos, solicitando o nome da View. Em seguida, alterações nas configurações devem ser salvas automaticamente a cada 30 segundos.	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Média
<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>	Alexandre Lenz	

<b>RN09 – Log out automático</b>		
<b>Descrição:</b>	O sistema deve encerrar a sessão de autenticação do usuário automaticamente após 5 minutos de inatividade.	
<b>Versão: 1.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Média
<b>Origem:</b>	Alexandre Lenz	
<b>Modificado por:</b>		

## 5. Verificação de Requisitos

O processo de verificação será feito através da inspeção do documento com o objetivo de avaliar se os requisitos estão sendo documentados de forma correta, além de identificar inconsistências entre eles:

- Verificar os conflitos de requisitos;
- Verificar a consistência dos requisitos;
- Verificar a completude dos requisitos;
- Verificar a existência de requisitos ambíguos.

## **6. Validação de Requisitos**

Nessa etapa o objetivo é avaliar se o que foi entregue corresponde às expectativas do cliente. Em outras palavras, se os requisitos estão sendo definidos com clareza o suficiente para atender às expectativas do cliente e se o sistema é realmente aquilo que o cliente quer. A validação final será realizada pelo próprio cliente, verificando se:

- O sistema possui as funções para suprir as necessidades dos usuários?
- Foram incluídas todas as funcionalidades requisitadas?
- Existe algum requisito conflitante?
- Todos requisitos estão descritos de forma clara e objetiva?
- Os requisitos podem ser verificados?
- Os requisitos são rastreáveis?

ANEXO B – Documento de casos de uso

	<p align="center"><b>FunRegulation</b>  <b>Construção e Manipulação de</b>  <b>Redes Regulatórias de Genes</b>  <b>de Fungos</b></p>	
	<p align="center"><b>Documento de Análise de</b>  <b>Requisitos</b></p>	

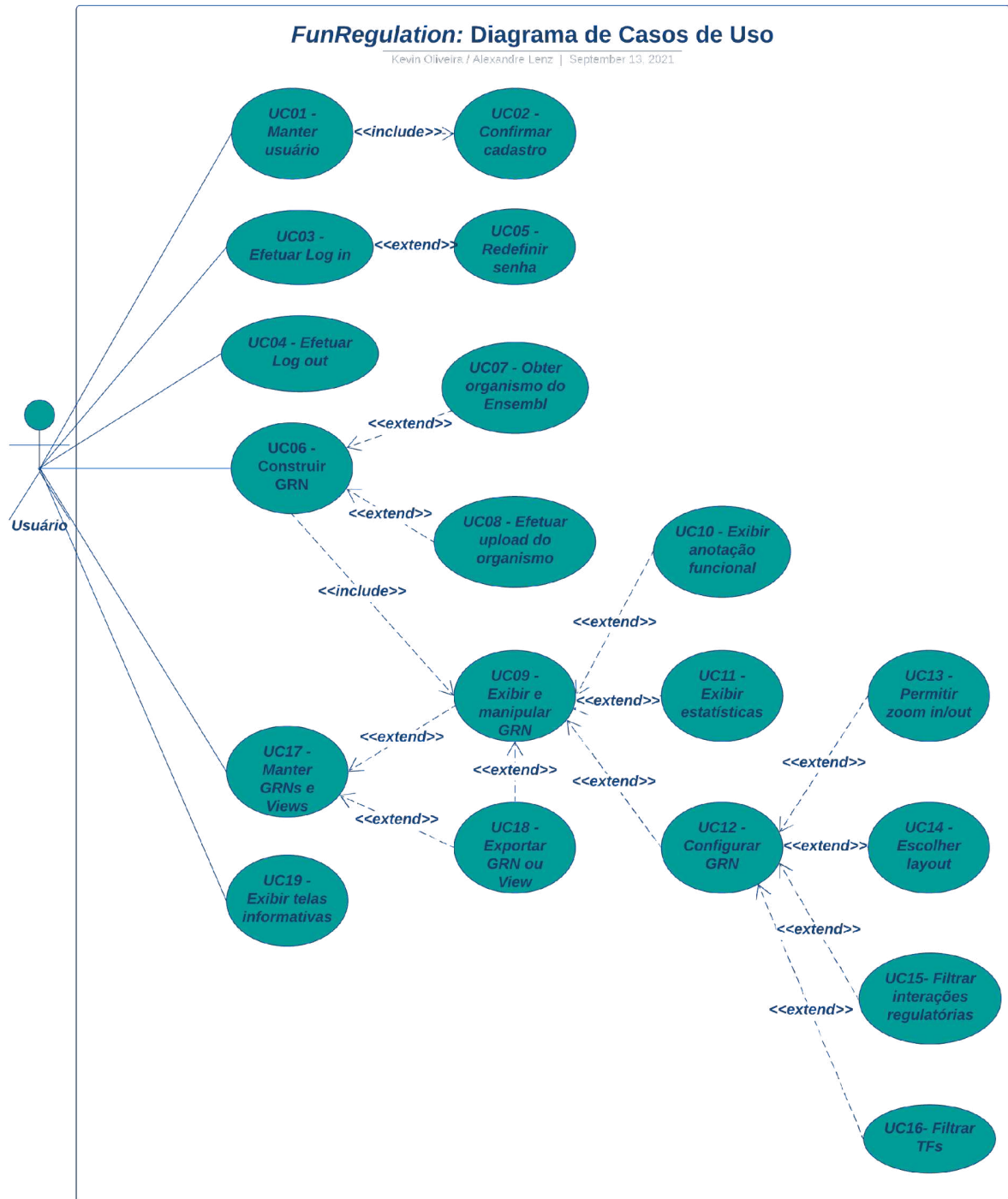
## Sumário

1. Introdução	1
2. Diagrama de Casos de Uso	2
3. Descrição de Casos de Uso	3

### 1. Introdução

Este documento tem por objetivo detalhar a análise dos requisitos do sistema FunRegulation, que compreende uma plataforma para construção e manipulação de redes regulatórias de genes de fungos. A análise dos requisitos deste sistema inclui o diagrama de casos de uso e as descrições detalhadas dos casos de uso.

## 2. Diagrama de Casos de Uso



### 3. Descrição de Casos de Uso

UC01 - Manter usuário		
<b>Subject:</b>	FunRegulation	
<b>Atores:</b>	Usuário	
<b>Objetivo:</b>	Cadastrar as informações do usuário para estarem habilitados a utilizar o sistema.	
<b>Modificado por:</b>	Alexandre Lenz	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Média
Fluxos de eventos		
<b>Pré-condições:</b>	O usuário precisa ter executado o passo 2.2 do Fluxo normal do UC19.	
<b>Fluxo normal:</b>	<ol style="list-style-type: none"> <li>1. O sistema apresenta formulário com campos para cadastro do usuário;</li> <li>2. O usuário fornece o nome completo;</li> <li>3. O usuário fornece e-mail institucional, conforme RN03;</li> <li>4. O usuário fornece senha, conforme RN02;</li> <li>5. O usuário fornece confirmação da senha, conforme RN02;</li> <li>6. O usuário fornece o nome da instituição do usuário;</li> <li>7. O usuário fornece o País do usuário;</li> <li>8. Se for um usuário brasileiro, informa o Estado.</li> <li>9. O usuário finaliza o cadastro;</li> <li>10. O sistema executa o caso de uso UC02;</li> </ol>	
<b>Fluxos alternativos:</b>	Não se aplica.	
<b>Fluxos de exceção:</b>	<p>3e. O e-mail do usuário não atende os requisitos da RN03.</p> <ol style="list-style-type: none"> <li>1. O sistema informa para o usuário inserir um novo e-mail institucional.</li> </ol> <p>4e. A senha do usuário não atende os requisitos da RN02</p> <ol style="list-style-type: none"> <li>1. O sistema informa para o usuário digitar uma nova senha.</li> </ol> <p>9e. O e-mail informado já estava cadastrado.</p> <ol style="list-style-type: none"> <li>1. O sistema mostra mensagem de erro e pergunta se usuário deseja recuperar senha;</li> <li>2. Se usuário escolhe a opção recuperar senha, executa UC05.</li> </ol> <p>9e. A confirmação da senha é diferente da senha informada.</p> <ol style="list-style-type: none"> <li>1. O sistema informa o usuário que os campos de senhas estão diferentes;</li> </ol>	

	<ol style="list-style-type: none"> <li>2. O usuário corrige os campos de senha;</li> <li>3. Retorna para o passo 10 do Fluxo normal.</li> </ol>
<b>Pós-condições:</b>	Usuário cadastrado no sistema.

<b>UC02 - Confirmar cadastro</b>		
<b>Subject:</b>	FunRegulation	
<b>Atores:</b>	Usuário	
<b>Objetivo:</b>	Confirmar o cadastro, evitando uso de e-mail inválido.	
<b>Modificado por:</b>	Alexandre Lenz	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Média
<b>Fluxos de eventos</b>		
<b>Pré-condições:</b>	O usuário precisa ter executado o UC01.	
<b>Fluxo normal:</b>	<ol style="list-style-type: none"> <li>1. O usuário recebe o e-mail de confirmação de cadastro;</li> <li>2. O usuário acessa o link para confirmação do cadastro;</li> <li>3. O sistema mostra tela de confirmação do cadastro;</li> <li>4. O usuário confirma o cadastro;</li> <li>5. O sistema confirma o cadastro e redireciona para tela de log in;</li> <li>6. O caso de uso é encerrado.</li> </ol>	
<b>Fluxos alternativos:</b>	1.a Usuário não recebeu o e-mail de confirmação de cadastro; <ol style="list-style-type: none"> <li>1. O usuário retorna ao sistema e tenta efetuar o log in;</li> </ol>	

	<ol style="list-style-type: none"> <li>2. O sistema informa que o cadastro não foi confirmado;</li> <li>3. O sistema possibilita o reenvio do e-mail de confirmação;</li> <li>4. O usuário solicita o reenvio do e-mail de confirmação;</li> <li>5. O sistema reenvia o e-mail;</li> <li>6. Retorna para o Fluxo normal.</li> </ol>
<b>Fluxos de exceção:</b>	Não se aplica.
<b>Pós-condições:</b>	Usuário habilitado para logar no sistema.

UC03 - Efetuar Log in		
<b>Subject:</b>	FunRegulation	
<b>Atores:</b>	Usuário	
<b>Objetivo:</b>	Realizar o login no sistema para ter acesso às funcionalidades de construção e manipulação de GRNs.	
<b>Modificado por:</b>	Alexandre Lenz	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Média
Fluxos de eventos		
<b>Pré-condições:</b>	Usuário habilitado para logar no sistema.	
<b>Fluxo normal:</b>	<ol style="list-style-type: none"> <li>1. O usuário acessa a opção Log in;</li> <li>2. O usuário informa o e-mail;</li> <li>3. O usuário informa a senha, conforme RN02;</li> <li>4. O usuário confirma;</li> <li>5. O sistema mostra tela inicial.</li> </ol>	
<b>Fluxos alternativos:</b>	Não se aplica.	
<b>Fluxos de exceção:</b>	<p>3e. O usuário esqueceu sua senha.</p> <ol style="list-style-type: none"> <li>1. O sistema executa o caso de uso UC05;</li> </ol> <p>4e. E-mail não cadastrado.</p> <ol style="list-style-type: none"> <li>1. O sistema mostra mensagem de erro informando que o e-mail informado não está cadastrado;</li> </ol> <p>4e. Senha incorreta</p> <ol style="list-style-type: none"> <li>1. O sistema mostra mensagem de erro informando que a senha está incorreta;</li> </ol>	

	<ol style="list-style-type: none"> <li>2. O usuário corrige a senha;</li> <li>3. Retorna ao passo 4 do Fluxo normal.</li> </ol>
<b>Pós-condições:</b>	Usuário autenticado no sistema.

UC04 - Efetuar Log out		
<b>Subject:</b>	FunRegulation	
<b>Atores:</b>	Usuário	
<b>Objetivo:</b>	O usuário finalizará seu acesso ao sistema.	
<b>Modificado por:</b>	Alexandre Lenz	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Média
Fluxos de eventos		
<b>Pré-condições:</b>	O usuário precisa estar logado no sistema.	
<b>Fluxo normal:</b>	<ol style="list-style-type: none"> <li>1. O usuário solicita o logout no sistema;</li> <li>2. O sistema apresenta mensagem solicitando confirmação;</li> <li>3. O usuário confirma a solicitação;</li> <li>4. O sistema desconecta o usuário e finaliza a sessão.</li> </ol>	
<b>Fluxos alternativos:</b>	<ol style="list-style-type: none"> <li>1a. O usuário permanece inativo no sistema, conforme RN09.               <ol style="list-style-type: none"> <li>1. Retorna ao passo 4 do Fluxo normal.</li> </ol> </li> <li>1a. O usuário fecha a janela do browser.               <ol style="list-style-type: none"> <li>1. Retorna ao passo 4 do Fluxo normal.</li> </ol> </li> <li>1a. O usuário utiliza o botão Voltar do browser e sai do sistema.               <ol style="list-style-type: none"> <li>1. Retorna ao passo 4 do Fluxo normal.</li> </ol> </li> <li>3a. O usuário cancela a solicitação de logout.               <ol style="list-style-type: none"> <li>1. O usuário permanece autenticado no sistema.</li> </ol> </li> </ol>	
<b>Fluxos de exceção:</b>	Não se aplica.	
<b>Pós-condições:</b>	O sistema encerra a sessão e o usuário é desconectado do sistema.	

UC05 - Redefinir senha		
<b>Subject:</b>	FunRegulation	
<b>Atores:</b>	Usuário	
<b>Objetivo:</b>	Caso o usuário redefinir sua senha, ele poderá solicitar a redefinição através do e-mail cadastrado.	
<b>Modificado por:</b>	Alexandre Lenz	
<b>Versão:</b> 2.0	<b>Status:</b> Aprovado	<b>Prioridade:</b> Média
Fluxos de eventos		
<b>Pré-condições:</b>	O usuário deve estar cadastrado no sistema e ter acesso ao e-mail cadastrado para permitir a redefinição da senha.	
<b>Fluxo normal:</b>	<ol style="list-style-type: none"> <li>1. O usuário solicita a redefinição de senha através do botão “Esqueci minha senha” na tela de Log in;</li> <li>2. O sistema solicita o e-mail do usuário para redefinir a senha;</li> <li>3. O usuário informa o e-mail;</li> <li>4. O usuário confirma a solicitação de redefinição de senha;</li> <li>5. O sistema envia um e-mail para o usuário, contendo um link para a tela de redefinição de senha;</li> <li>6. O usuário acessa o link, redirecionado para a tela de redefinição de senha;</li> <li>7. O usuário preenche os campos “Nova senha” e “Repetir nova senha”, conforme RN02;</li> <li>8. O usuário confirma a redefinição de senha;</li> <li>9. O sistema confirma a nova senha;</li> <li>10. O sistema executa o caso de uso UC03;</li> </ol>	
<b>Fluxos alternativos:</b>	Não se aplica.	
<b>Fluxos de exceção:</b>	<p>8e. A senha do usuário não atende os requisitos da RN02</p> <ol style="list-style-type: none"> <li>1. O sistema informa para o usuário digitar uma nova senha.</li> <li>2. Retorna para o passo 8 do Fluxo normal.</li> </ol> <p>8e. A confirmação da senha é diferente da senha informada.</p> <ol style="list-style-type: none"> <li>4. O sistema informa o usuário que os campos de senhas estão diferentes;</li> <li>5. O usuário corrige os campos de senha;</li> <li>6. Retorna para o passo 8 do Fluxo normal.</li> </ol>	
<b>Pós-condições:</b>	A senha do usuário foi redefinida, permitindo Log in no sistema.	

UC06 - Construir GRN		
<b>Subject:</b>	FunRegulation	
<b>Atores:</b>	Usuário	
<b>Objetivo:</b>	Este caso de uso compreende a funcionalidade mais importante do sistema, permitindo ao usuário a construção de um GRN a partir dos dados genômicos de um fungo.	
<b>Modificado por:</b>	Alexandre Lenz	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Alta
Fluxos de eventos		
<b>Pré-condições:</b>	O usuário precisa estar logado no sistema.	
<b>Fluxo normal:</b>	<ol style="list-style-type: none"> <li>1. O sistema deve permitir a escolha de um organismo para construir a GRN a partir de uma das duas opções:               <ol style="list-style-type: none"> <li>1.1. O usuário pode escolher um organismo obtido do Ensembl Fungi, executando UC07; ou</li> <li>1.2. O usuário pode efetuar upload de arquivos genômicos de um organismo, executando UC08.</li> </ol> </li> <li>2. O usuário deve selecionar quais interações regulatórias devem ser consideradas para construção da GRN:               <ol style="list-style-type: none"> <li>2.1. “Somente ortologia de organismo modelo”; ou</li> <li>2.2. “Ortologia de organismo modelo e sítios de ligação dos fatores de transcrição”.</li> </ol> </li> <li>3. O usuário aciona a construção da GRN;</li> <li>4. O sistema deve construir a GRN com os dados obtidos nos passos seguintes, mostrando um workflow com: passos executados, em execução e a executar. O sistema também deve mostrar uma barra de progresso e estimativa de tempo para construção da GRN;               <ol style="list-style-type: none"> <li>4.1. Quando a opção “Somente ortologia de organismo modelo” for selecionada, o sistema deve:                   <ol style="list-style-type: none"> <li>4.1.1. Identificar os genes ortólogos utilizando o software ProteinOrtho;</li> <li>4.1.2. Carregar as relações comprovadas entre TFs e TGs do organismo modelo;</li> </ol> </li> </ol> </li> </ol>	

	<p>4.1.3. Quando são encontrados os ortólogos para TF e TG de uma interação do organismo modelo, uma nova interação TF-TG é criada para o organismo alvo.</p> <p>4.2. Quando a opção “Ortologia de organismo modelo e sítios de ligação dos fatores de transcrição” for selecionada, o sistema deve:</p> <p>4.2.1. Executar os passos 5.1.1 a 5.1.3;</p> <p>4.2.2. Extrair os promotores 1000bp upstream de cada TG do genoma do organismo alvo;</p> <p>4.2.3. Obter as matrizes PWM do banco de dados CIS-BP;</p> <p>4.2.4. Realizar a predição dos sítios de ligação dos fatores de transcrição utilizando web service RSAT e os promotores e PWMs do passo anterior;</p> <p>4.2.5. Compilar os resultados do RSAT;</p> <p>5. O sistema finaliza a construção e salva a GRN na lista de GRNs do usuário;</p> <p>6. O sistema executa o UC09.</p>
<b>Fluxos alternativos:</b>	<b>As alternativas foram descritas no fluxo normal para facilitar a compreensão.</b>
<b>Fluxos de exceção:</b>	<p>1e. Caso o usuário não esteja logado, o sistema redireciona para o UC03;</p> <p>5e. Ocorreu algum erro durante a construção da GRN;</p> <ol style="list-style-type: none"> <li>1. O sistema deve mostrar uma mensagem ao usuário informando o erro que ocorreu;</li> <li>2. O sistema deve armazenar registro de Log e enviar e-mail, conforme RNF04;</li> </ol>
<b>Pós-condições:</b>	A GRN é construída e armazenada.

<b>UC07 - Obter organismo do Ensembl</b>		
<b>Subject:</b>	FunRegulation	
<b>Atores:</b>	Usuário	
<b>Objetivo:</b>	Fornecer uma lista de organismos obtidos através da plataforma Ensembl Fungi para que o usuário possa escolher o organismo que deseja analisar.	
<b>Modificado por:</b>	Alexandre Lenz	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Média

Fluxos de eventos	
<b>Pré-condições:</b>	O usuário executou o passo 1 do Fluxo normal do UC06.
<b>Fluxo normal:</b>	<ol style="list-style-type: none"> <li>1. O sistema disponibiliza uma lista de organismos obtida da plataforma Ensembl Fungi, conforme RN04 e RNF01;</li> <li>2. O sistema mostra uma breve descrição de cada organismo, a partir da função mouse-over;</li> <li>3. O usuário seleciona o organismo desejado;</li> <li>4. O sistema retorna ao passo 3 do Fluxo normal do UC06.</li> </ol>
<b>Fluxos alternativos:</b>	Não se aplica.
<b>Fluxos de exceção:</b>	<p>3e. O usuário não encontrou o organismo que desejava analisar.</p> <ol style="list-style-type: none"> <li>1. O usuário pode efetuar upload de arquivos de um organismo, executando o UC08.</li> </ol>
<b>Pós-condições:</b>	Os dados genômicos do organismo são carregados no sistema para construção da GRN.

UC08 - Efetuar upload do organismo		
<b>Subject:</b>	FunRegulation	
<b>Atores:</b>	Usuário	
<b>Objetivo:</b>	Carregar arquivos genômicos de organismos que não estejam catalogados pela plataforma Ensembl Fungi.	
<b>Modificado por:</b>	Alexandre Lenz	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Alta
Fluxos de eventos		
<b>Pré-condições:</b>	O usuário executou o passo 1 do Fluxo normal do UC06.	
<b>Fluxo normal:</b>	<ol style="list-style-type: none"> <li>1. O usuário informa o gênero, espécie e linhagem do organismo que deseja analisar;</li> <li>2. O sistema valida o gênero, conforme RN05;</li> <li>3. O sistema habilita a possibilidade de upload de arquivos via botão de seleção de arquivos ou drag-and-drop;</li> <li>4. O usuário seleciona os arquivos e efetua o upload;</li> <li>5. O sistema valida os arquivos conforme RN05;</li> </ol>	

	6. O sistema retorna ao passo 3 do Fluxo normal do UC06.
<b>Fluxos alternativos:</b>	Não se aplica.
<b>Fluxos de exceção:</b>	<p>2e. O gênero informado pelo usuário não atende à RN05.</p> <ol style="list-style-type: none"> <li>1. O sistema mostra mensagem indicando que a construção da GRN somente deve ser efetuada para organismos das 4 ordens: <i>Eurotiales</i>, <i>Sordarialises</i>, <i>Saccharomycetales</i> e <i>Hypocreales</i>.</li> </ol> <p>5e. O usuário não fez upload dos arquivos obrigatórios.</p> <ol style="list-style-type: none"> <li>1. O sistema mostra mensagem indicando quais são os arquivos obrigatórios.</li> </ol> <p>5e. A validação dos arquivos identificou incompatibilidade nos formatos dos arquivos.</p> <ol style="list-style-type: none"> <li>1. O sistema mostra mensagem indicando quais arquivos não estão no formato correto.</li> </ol>
<b>Pós-condições:</b>	Os dados genômicos do organismo são carregados no sistema para construção da GRN.

UC09 - Exibir e manipular GRN		
<b>Subject:</b>	FunRegulation	
<b>Atores:</b>	Usuário	
<b>Objetivo:</b>	Essa funcionalidade compreende a exibição gráfica da GRN e a manipular da mesma a partir de uma série de configurações.	
<b>Modificado por:</b>	Alexandre Lenz	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Alta
Fluxos de eventos		
<b>Pré-condições:</b>	O usuário precisa ter executado o UC06 ou UC17.	
<b>Fluxo normal:</b>	<ol style="list-style-type: none"> <li>1. O sistema exibe tela com a representação gráfica da GRN, conforme RN06 e RN07;</li> <li>2. O sistema permite uma série de ações a partir da tela de visualização da GRN: <ol style="list-style-type: none"> <li>2.1. O usuário seleciona um nó da GRN para exibir anotação funcional e o sistema executa o UC10;</li> </ol> </li> </ol>	

	<p>2.2. O usuário deseja exibir as estatísticas gerais da GRN e o sistema executa o UC11;</p> <p>2.3. O usuário deseja configurar a visualização da GRN e o sistema executa o UC12;</p> <p>2.4. O usuário deseja exportar a GRN e o sistema executa o UC18;</p> <p>2.5. O usuário deseja salvar uma nova View e o sistema executa o UC17;</p> <p>3. O caso de uso é encerrado.</p>
<b>Fluxos alternativos:</b>	Não se aplica
<b>Fluxos de exceção:</b>	Não se aplica.
<b>Pós-condições:</b>	GRN exibida na tela.

<b>UC10 - Exibir anotação funcional</b>		
<b>Subject:</b>	FunRegulation	
<b>Atores:</b>	Usuário	
<b>Objetivo:</b>	Mostrar anotação funcional referente a cada gene da GRN.	
<b>Modificado por:</b>	Alexandre Lenz	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Média
<b>Fluxos de eventos</b>		
<b>Pré-condições:</b>	O usuário deve estar executando o caso UC09.	
<b>Fluxo normal:</b>	<p>1. O usuário clica com botão esquerdo em um gene da GRN;</p> <p>2. O sistema obtém as informações de anotação funcional de acordo com a seleção do organismo nos passos 2.1 ou 2.2 do Fluxo normal do UC06.</p> <p>3. O sistema exibe as informações de anotação funcional do gene selecionado em uma janela modal;</p> <p>4. O usuário fecha a janela modal;</p> <p>5. O caso de uso é encerrado.</p>	
<b>Fluxos alternativos:</b>	Não se aplica.	

<b>Fluxos de exceção:</b>	Não se aplica.
<b>Pós-condições:</b>	Uma janela modal com as informações de anotação funcional é mostrada ao usuário.

<b>UC11 - Exibir estatísticas</b>		
<b>Subject:</b>	FunRegulation	
<b>Atores:</b>	Usuário	
<b>Objetivo:</b>	Exibir estatísticas gerais da GRN	
<b>Modificado por:</b>	Alexandre Lenz	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Média
<b>Fluxos de eventos</b>		
<b>Pré-condições:</b>	O usuário deve estar executando o caso UC09.	
<b>Fluxo normal:</b>	<ol style="list-style-type: none"> <li>1. O usuário clica no botão Estatísticas;</li> <li>2. O sistema calcula as estatísticas gerais da GRN ou da View, de acordo com medidas de centralidade.</li> <li>3. O sistema exibe as estatísticas gerais da GRN ou da View em uma janela modal;</li> <li>4. O usuário fecha a janela modal;</li> <li>5. O caso de uso é encerrado.</li> </ol>	
<b>Fluxos alternativos:</b>	Não se aplica.	
<b>Fluxos de exceção:</b>	Não se aplica.	
<b>Pós-condições:</b>	O usuário visualiza as estatísticas gerais da GRN ou da View.	

<b>UC12 - Configurar GRN</b>	
<b>Subject:</b>	FunRegulation

<b>Atores:</b>	Usuário	
<b>Objetivo:</b>	Exibir estatísticas gerais da GRN	
<b>Modificado por:</b>	Alexandre Lenz	
<b>Versão: 1.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Baixa
<b>Fluxos de eventos</b>		
<b>Pré-condições:</b>	O usuário deve estar executando o caso UC09.	
<b>Fluxo normal:</b>	<ol style="list-style-type: none"> <li>1. O sistema possibilita uma série de configurações a partir da tela de visualização da GRN: <ol style="list-style-type: none"> <li>1.1. O usuário deseja ampliar a GRN e executa o UC13;</li> <li>1.2. O usuário deseja mudar o layout da GRN e executa o UC14;</li> <li>1.3. O usuário deseja filtrar as interações regulatórias mostradas na GRN e executa o UC15;</li> <li>1.4. O usuário deseja filtrar os fatores de transcrições mostrados na GRN e executa o UC16;</li> </ol> </li> <li>2. O sistema salva automaticamente as configurações realizadas conforme RN08. <ol style="list-style-type: none"> <li>2.1. O sistema mostra um campo para que o usuário informe o nome da View para salvar as configurações realizadas;</li> <li>2.2. O usuário informa o nome da View;</li> <li>2.3. O usuário confirma e o sistema salva a nova View.</li> </ol> </li> <li>3. O caso de uso é encerrado.</li> </ol>	
<b>Fluxos alternativos:</b>	2.a Caso as configurações tenham sido realizadas para uma View previamente salva, o sistema salva automaticamente as novas configurações de maneira transparente para o usuário, sem necessidade de informar um nome para a View.	
<b>Fluxos de exceção:</b>	Não se aplica.	
<b>Pós-condições:</b>	Configurações da GRN realizadas e salvas como uma View.	

### UC13 - Permitir zoom in/out

<b>Subject:</b>	FunRegulation
-----------------	---------------

<b>Atores:</b>	Usuário	
<b>Objetivo:</b>	Permitir que os usuários modifiquem a visualização da rede a partir de uma ferramenta de Zoom in/out.	
<b>Modificado por:</b>	Leandro Lopes	
<b>Versão: 1.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Baixa
<b>Fluxos de eventos</b>		
<b>Pré-condições:</b>	O usuário deve estar executando o caso UC12.	
<b>Fluxo normal:</b>	<ol style="list-style-type: none"> <li>1. O sistema disponibiliza a ferramenta de Zoom. A funcionalidade deve ser acessível por: <ol style="list-style-type: none"> <li>1.1. Scroll do mouse;</li> <li>1.2. Pinça no touchpad;</li> <li>1.3. teclas de atalho + e -;</li> <li>1.4. botões +, - e fit-to-window.</li> </ol> </li> <li>2. O usuário amplia ou reduz a visualização gráfica da GRN a partir de um dos comandos de Zoom;</li> <li>3. O sistema amplia ou reduz a visualização gráfica da GRN.</li> <li>4. O sistema retorna ao passo 2 do Fluxo normal do UC12.</li> </ol>	
<b>Fluxos alternativos:</b>	Não se aplica.	
<b>Fluxos de exceção:</b>	Não se aplica.	
<b>Pós-condições:</b>	A visualização da GRN é ajustada.	

<b>UC14 - Escolher layout</b>		
<b>Subject:</b>	FunRegulation	
<b>Atores:</b>	Usuário	
<b>Objetivo:</b>	Possibilitar a exibição do layout da GRN conforme a preferência do usuário.	
<b>Modificado por:</b>	Alexandre Lenz	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Média

Fluxos de eventos	
<b>Pré-condições:</b>	O usuário deve estar executando o caso UC12.
<b>Fluxo normal:</b>	<ol style="list-style-type: none"> <li>1. O sistema disponibiliza uma lista de layouts de exibição predefinidos: <ol style="list-style-type: none"> <li>1.1. <i>grid</i></li> <li>1.2. <i>circle</i></li> <li>1.3. <i>concentric</i></li> <li>1.4. <i>avsd</i></li> <li>1.5. <i>dagre</i></li> <li>1.6. <i>breadthfirst</i></li> <li>1.7. <i>elk</i></li> <li>1.8. <i>klay</i></li> <li>1.9. <i>fcose</i></li> <li>1.10. <i>cola</i></li> <li>1.11. <i>cise</i></li> <li>1.12. <i>euler</i></li> <li>1.13. <i>spread</i></li> </ol> </li> <li>2. O usuário seleciona um layout de exibição predefinido;</li> <li>3. O sistema aplica o layout selecionado na View;</li> <li>4. O sistema retorna ao passo 2 do Fluxo normal do UC12.</li> </ol>
<b>Fluxos alternativos:</b>	Não se aplica.
<b>Fluxos de exceção:</b>	Não se aplica.
<b>Pós-condições:</b>	O layout da GRN é alterado.

UC15 - Filtrar interações regulatórias		
<b>Subject:</b>	FunRegulation	
<b>Atores:</b>	Usuário	
<b>Objetivo:</b>	Filtrar as arestas que devem ser mostradas na GRN.	
<b>Modificado por:</b>	Alexandre Lenz	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Muito baixa
Fluxos de eventos		

<b>Pré-condições:</b>	O usuário deve estar executando o caso UC12.
<b>Fluxo normal:</b>	<ol style="list-style-type: none"> <li>O sistema permite o filtro das interações regulatórias que são apresentadas na GRN. O filtro deve ser acionado a partir de um botão do tipo On-Off: <ol style="list-style-type: none"> <li>Somente Ortologia (Off) - Protein-DNA e Ortologia (On);</li> <li>A opção <i>default</i> é Protein-DNA e Ortologia (On);</li> </ol> </li> <li>O usuário altera o filtro das interações regulatórias;</li> <li>O sistema aplica o filtro selecionado na View, conforme RN07;</li> <li>O sistema retorna ao passo 2 do Fluxo normal do UC12.</li> </ol>
<b>Fluxos alternativos:</b>	Não se aplica.
<b>Fluxos de exceção:</b>	<ol style="list-style-type: none"> <li>Caso no passo 2.1 do Fluxo normal do UC06 tenha sido escolhida a opção "Somente ortologia de organismo modelo". <ol style="list-style-type: none"> <li>O botão On-Off fica desabilitado na opção Somente Ortologia (Off).</li> </ol> </li> </ol>
<b>Pós-condições:</b>	As interações regulatórias mostradas na GRN são filtradas.

UC16 - Filtrar TFs		
<b>Subject:</b>	FunRegulation	
<b>Atores:</b>	Usuário	
<b>Objetivo:</b>	Filtrar os fatores de transcrição que devem ser mostradas na GRN.	
<b>Modificado por:</b>	Alexandre Lenz	
<b>Versão:</b> 1.0	<b>Status:</b> Aprovado	<b>Prioridade:</b> Média
Fluxos de eventos		
<b>Pré-condições:</b>	O usuário deve estar executando o caso UC12.	
<b>Fluxo normal:</b>	<ol style="list-style-type: none"> <li>O sistema permite o filtro dos fatores de transcrição (TFs) que são apresentadas na GRN. O filtro deve ser acionado a partir de um campo de texto. <ol style="list-style-type: none"> <li>O usuário pode informar até 10 TFs (locus tag e/ou gene symbol) separadas por “;”</li> <li>O campo de texto deve ser mostrado com um exemplo de preenchimento (FOXG_01234;CreA);</li> </ol> </li> </ol>	

	<ol style="list-style-type: none"> <li>2. O usuário informa os TFs e aciona a opção Filtrar TFs;</li> <li>3. O sistema aplica o filtro de TFs na View, são mostrados apenas os TFs informados e os TGs regulados pelos TFs informados;</li> <li>4. O sistema retorna ao passo 2 do Fluxo normal do UC12.</li> </ol>
<b>Fluxos alternativos:</b>	Não se aplica.
<b>Fluxos de exceção:</b>	<ol style="list-style-type: none"> <li>1e. Caso o usuário informe um locus tag ou gene symbol que não é um TF ou que não tenha sido encontrado na GRN; <ol style="list-style-type: none"> <li>1. O sistema mostra mensagem de erro indicando que o(s) TF(s) informado(s) não foi(ram) encontrado(s).</li> </ol> </li> </ol>
<b>Pós-condições:</b>	Apenas os TFs informados e seus respectivos TGs são mostrados na GRN.

<b>UC17 - Manter GRNs e Views</b>		
<b>Subject:</b>	FunRegulation	
<b>Atores:</b>	Usuário	
<b>Objetivo:</b>	Manter uma lista das GRNs criadas por um usuário e das Views associadas a cada GRN.	
<b>Modificado por:</b>	Alexandre Lenz	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Alta
<b>Fluxos de eventos</b>		
<b>Pré-condições:</b>	O usuário precisa estar logado no sistema.	
<b>Fluxo normal:</b>	<ol style="list-style-type: none"> <li>1. O usuário acessa a opção My GRNs;</li> <li>2. O sistema mostra a lista de GRNs criadas pelo usuário e, de maneira hierárquica, mostra as Views salvas para cada GRN; <ol style="list-style-type: none"> <li>2.1. Cada GRN da lista possui os seguintes botões: <ol style="list-style-type: none"> <li>2.1.1. Renomear - Fluxo Alternativo 2.1.1a.</li> <li>2.1.2. Visualizar - executa o UC09;</li> <li>2.1.3. Excluir - Fluxo Alternativo 2.1.3a.</li> <li>2.1.4. Exportar - Fluxo Alternativo 2.1.4a.</li> </ol> </li> <li>2.2. Cada View da lista possui os seguintes botões: <ol style="list-style-type: none"> <li>2.2.1. Renomear - Fluxo Alternativo 2.2.1a.</li> <li>2.2.2. Visualizar - executa o UC09;</li> <li>2.2.3. Excluir - Fluxo Alternativo 2.2.3a.</li> </ol> </li> </ol> </li> </ol>	

	3. O caso de uso é encerrado.
<b>Fluxos alternativos:</b>	<p>1a O usuário deseja salvar uma nova View, a partir do passo 2.5 do Fluxo normal do UC09.</p> <ol style="list-style-type: none"> <li>1.1. O sistema mostra um campo para que o usuário informe o nome da <i>View</i> para salvar as configurações realizadas;</li> <li>1.2. O usuário informa o nome da <i>View</i>;</li> <li>1.3. O usuário confirma e o sistema salva a nova <i>View</i>.</li> </ol> <p>2.1.1a e 2.2.1a O usuário deseja renomear uma GRN ou View;</p> <ol style="list-style-type: none"> <li>1. O sistema solicita que o usuário altere o nome da GRN ou da View;</li> <li>2. O usuário altera o nome da GRN ou da View e confirma;</li> <li>3. O sistema atualiza o nome da GRN ou da View;</li> </ol> <p>2.1.3a O usuário solicita a exclusão de uma GRN;</p> <ol style="list-style-type: none"> <li>1. O sistema solicita confirmação da exclusão da GRN e de todas as <i>Views</i> associadas; <ol style="list-style-type: none"> <li>1.1. O usuário confirma a exclusão da GRN e de todas as <i>Views</i> associadas;</li> <li>1.2. O sistema exclui a GRN e todas as <i>Views</i> associadas;</li> </ol> </li> </ol> <p>2.1.4a O usuário solicita a exportação de uma GRN;</p> <ol style="list-style-type: none"> <li>1. O sistema cria um arquivo compactado contendo: <ol style="list-style-type: none"> <li>1.1. Arquivo TSV com a GRN completa;</li> <li>1.2. Arquivo TSV com a identificação de genes ortólogos;</li> <li>1.3. Arquivo TSV com os resultados da predição de sítios de ligação dos fatores de transcrição;</li> </ol> </li> <li>2. O download do arquivo compactado é realizado.</li> </ol> <p>2.2.3a O usuário solicita a exclusão de uma <i>View</i>;</p> <ol style="list-style-type: none"> <li>1. O sistema solicita confirmação da exclusão da <i>View</i>; <ol style="list-style-type: none"> <li>1.1. O usuário confirma a exclusão da <i>View</i>;</li> <li>1.2. O sistema exclui a <i>View</i>;</li> </ol> </li> </ol>
<b>Fluxos de exceção:</b>	Não se aplica.
<b>Pós-condições:</b>	GRNs e suas respectivas Views são armazenadas e mantidas no sistema de maneira hierárquica.

## UC18 - Exportar GRN ou View

<b>Subject:</b>	FunRegulation	
<b>Atores:</b>	Usuário	
<b>Objetivo:</b>	Permitir que o usuário exporte a GRN em diferentes formatos para uso em outros softwares de análise, relatórios, artigos ou outros fins.	
<b>Modificado por:</b>	Alexandre Lenz	
<b>Versão: 2.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Alta
<b>Fluxos de eventos</b>		
<b>Pré-condições:</b>	O usuário deve estar executando o caso UC09.	
<b>Fluxo normal:</b>	<ol style="list-style-type: none"> <li>1. O sistema permite a exportação da GRN ou de uma View a partir de um botão Export;</li> <li>2. O usuário acessa a opção Export e escolhe um formato para exportação: <ol style="list-style-type: none"> <li>2.1. TSV (Tab-Separated Values);</li> <li>2.2. PNG (Portable Network Graphics);</li> <li>2.3. SVG (Scalable Vector Graphics);</li> <li>2.4. PDF (Portable Document Format).</li> </ol> </li> <li>3. O sistema exporta a GRN ou a View no formato selecionado e o download do arquivo é realizado.</li> </ol>	
<b>Fluxos alternativos:</b>	<ol style="list-style-type: none"> <li>2.1a O usuário solicita a exportação da GRN no formato TSV; <ol style="list-style-type: none"> <li>1. O sistema cria um arquivo compactado contendo: <ol style="list-style-type: none"> <li>1.1. Arquivo TSV com a GRN completa;</li> <li>1.2. Arquivo TSV com a identificação de genes ortólogos;</li> <li>1.3. Arquivo TSV com os resultados da predição de sítios de ligação dos fatores de transcrição;</li> </ol> </li> <li>2. Retorna ao passo 3 do Fluxo normal.</li> </ol> </li> <li>2.2a O usuário solicita a exportação da GRN no formato PNG; <ol style="list-style-type: none"> <li>1. O sistema permite a escolha da Resolução com valores entre 100 e 500 dpi;</li> <li>2. O usuário informa a resolução desejada e confirma a exportação;</li> <li>3. Retorna ao passo 3 do Fluxo normal.</li> </ol> </li> </ol>	
<b>Fluxos de exceção:</b>	Não se aplica.	
<b>Pós-condições:</b>	O usuário poderá utilizar a rede no formato desejado externamente ao sistema.	

UC19 - Exibir telas informativas		
<b>Subject:</b>	FunRegulation	
<b>Atores:</b>	Usuário	
<b>Objetivo:</b>	Mostrar telas informativas ao usuário contendo página inicial, tutorial e contato.	
<b>Modificado por:</b>	Alexandre Lenz	
<b>Versão: 1.0</b>	<b>Status:</b> Aprovado	<b>Prioridade:</b> Baixa
Fluxos de eventos		
<b>Pré-condições:</b>	Não se aplica.	
<b>Fluxo normal:</b>	<ol style="list-style-type: none"> <li>1. O usuário acessa a URL da aplicação;</li> <li>2. O sistema apresenta a tela inicial para o usuário contendo as opções: <ol style="list-style-type: none"> <li>2.1. Sign in - executa o UC03;</li> <li>2.2. Sign up - executa o UC01;</li> <li>2.3. Log out (somente após executar UC03) - executa o UC04;</li> <li>2.4. Construct GRN - executa o UC06;</li> <li>2.5. My GRNs (somente após executar UC03)- executa o UC17;</li> <li>2.6. Explore;</li> <li>2.7. Contact;</li> </ol> </li> <li>3. O caso de uso é encerrado.</li> </ol>	
<b>Fluxos alternativos:</b>	<p>2.6a O usuário seleciona a opção “Explore”;</p> <ol style="list-style-type: none"> <li>1. O sistema inicia o tutorial demonstrando o processo de funcionamento da ferramenta: <ol style="list-style-type: none"> <li>1.1. Dados/Arquivos de entrada; <ol style="list-style-type: none"> <li>1.1.1. Exemplos de arquivos de entrada; <ol style="list-style-type: none"> <li>1.1.1.1. Genoma (fasta)</li> <li>1.1.1.2. Anotação de genes (gff3)</li> <li>1.1.1.3. Proteínas (fasta)</li> <li>1.1.1.4. Anotação funcional (tsv)</li> </ol> </li> <li>1.2. Resultados esperados; <ol style="list-style-type: none"> <li>1.2.1. Exemplos de filtros e layouts;</li> </ol> </li> <li>1.3. Como analisar resultados;</li> </ol> </li> </ol> </li> </ol> <p>2.7a O usuário seleciona a opção “Contact”;</p> <ol style="list-style-type: none"> <li>1. O sistema mostra informações de contato: <ol style="list-style-type: none"> <li>1.1. E-mail para problemas científicos;</li> </ol> </li> </ol>	

	1.2. E-mail para problemas técnicos; 1.3. Citação do artigo publicado.
<b>Fluxos de exceção:</b>	Não se aplica.
<b>Pós-condições:</b>	O usuário capacitado a utilizar o sistema.