



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

SAMUEL SANTOS VIEIRA

**FUNREGULATION: CONSTRUÇÃO DE UMA APLICAÇÃO WEB (*FRONT-END*)
PARA VISUALIZAÇÃO E MANIPULAÇÃO DE REDES REGULATÓRIAS DE
GENES PARA FUNGOS**

**SALVADOR, BAHIA, BRASIL
2023**

SAMUEL SANTOS VIEIRA

FUNREGULATION: CONSTRUÇÃO DE UMA APLICAÇÃO WEB (*FRONT-END*) PARA
VISUALIZAÇÃO E MANIPULAÇÃO DE REDES REGULATÓRIAS DE GENES PARA
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 parcial à obtenção do grau de bacharel em Sistemas de Informação. Área de Concentração: Ciência da Computação

Orientador: Alexandre Rafael Lenz

SALVADOR, BAHIA, BRASIL

2023

SAMUEL SANTOS VIEIRA

FUNREGULATION: CONSTRUÇÃO DE UMA APLICAÇÃO WEB (*FRONT-END*) PARA
VISUALIZAÇÃO E MANIPULAÇÃO DE REDES REGULATÓRIAS DE GENES PARA
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 parcial 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

Vagner de Souza Fonseca
Universidade do Estado da Bahia - UNEB

Diego Gervasio Frías Suárez
Universidade do Estado da Bahia - UNEB

Termo de Anuênciâ do Orientador

Declaro para os devidos fins que li e revisei este trabalho e atesto sua qualidade como resultado final desta monografia. Confirmo 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, por me capacitar e permitir que eu chegassem até aqui. A minha família e amigos, que me apoiaram durante toda a graduação. Aos meus professores, que contribuíram para minha formação pessoal e profissional. Ao meu orientador, o professor Alexandre Rafael Lenz, por ter me apoiado e contribuído durante todo o processo de desenvolvimento deste trabalho.

RESUMO

Um dos maiores desafios da biologia molecular é identificar as relações regulatórias entre genes reguladores da transcrição e seus alvos, pois a partir disso é possível compreender os fenômenos biológicos como, por exemplo, o crescimento e divisão celular. Através das Redes Regulatórias de Genes (GRN, do inglês *Gene Regulatory Network*) é possível entender qual é a influência de cada interação entre genes em um determinado organismo. Com esses dados, os pesquisadores podem obter avanços nos estudos científicos com um menor custo financeiro, já que o sequenciamento genético é um processo de alto custo. O objetivo deste trabalho é desenvolver uma solução de software para a visualização e manipulação de uma GRN. Essa solução contempla o *front-end* de uma aplicação Web, sendo responsável por permitir a visualização e a configuração das características da GRN de acordo com as necessidades do usuário. Para validar o funcionamento do sistema, foram construídas duas GRNs a partir dos fungos *Madurella mycetomatis* e *Cordyceps militaris*. Também foram realizados testes para apurar se o carregamento da GRN é eficiente em termos de tempo e memória. A aplicação Web *front-end FunRegulation*, para visualização e manipulação de GRNs de fungos contribui para que seja possível entender a relação entre genes nesses seres vivos. O projeto da jornada do usuário contribuiu para a decisão da disposição das funcionalidades construídas para o sistema. A realização dos testes de carregamento das duas GRNs geradas demonstrou a funcionalidade da arquitetura para visualização de redes construída com o Cytoscape.js, e também mostrou a eficiência de carregamento das GRNs em termos de tempo e memória. Considera-se que todos os objetivos propostos para a elaboração deste trabalho foram alcançados.

Palavras-chave: Bioinformática. Fungos. Redes Regulatórias de Genes.

ABSTRACT

One of the biggest challenges in molecular biology is to identify the regulatory relationships between transcription regulatory genes and their targets, as this makes it possible to understand biological phenomena such as, for example, cell growth and division. Through Gene Regulatory Networks (GRN) it is possible to understand the influence of each interaction between genes in a given organism. With this data, researchers can make advances in scientific studies at a lower financial cost, as genetic sequencing is a high-cost process. The objective of this work is to develop a software solution for visualizing and manipulating a GRN. This solution includes the front-end of a Web application, being responsible for allowing the visualization and configuration of GRN characteristics according to the user's needs. To validate the functioning of the system, two GRNs were constructed from the fungi *Madurella mycetomatis* and *Cordyceps militaris*. Tests were also carried out to determine whether loading the GRN is efficient in terms of time and memory. The FunRegulation front-end web application, for visualizing and manipulating fungal GRNs, helps to understand the relationship between genes in these living beings. The design of the user journey contributed to the decision on the layout of the functionalities built for the system. Carrying out loading tests on the two GRNs generated demonstrated the functionality of the network visualization architecture built with Cytoscape.js. Also, it showed the loading efficiency of GRNs in terms of time and memory. It is considered that all objectives proposed for the preparation of this work were achieved.

LISTA DE FIGURAS

Figura 1 – Exemplo de uma GRN. Fonte: o autor.	13
Figura 2 – Recursos que podem ser utilizados para inferir uma GRN. Fonte: (1)	13
Figura 3 – Exemplo do conceito de ortologia. Fonte: (2)	14
Figura 4 – Processo de construção de uma matriz de posição. Fonte: (3)	15
Figura 5 – Possíveis aplicações dos conhecimentos sobre fungos. Fonte: (4)	16
Figura 6 – Exemplo de aplicação do Cytoscape.js para visualização de grafos. Fonte: (5) .	19
Figura 7 – Modelo <i>Design Science Research</i> para o projeto. Fonte: o autor, baseado em Pimentel (2020) (6).	25
Figura 8 – Metodologia Scrum adaptada para o FunRegulation. Fonte: o autor, adaptada de (7)	27
Figura 9 – Jornada do usuário para o FunRegulation. Fonte: o autor.	30
Figura 10 – Arquitetura do <i>front-end</i> do FunRegulation. Fonte: o autor.	32
Figura 11 – Exemplo de retorno da requisição de organismos via API. Fonte: o autor. . .	33
Figura 12 – Exemplo de requisição de construção da GRN via API. Fonte: o autor. . .	34
Figura 13 – Exemplo de retorno da requisição de construção da GRN via API. Fonte: o autor.	34
Figura 14 – Exemplo de retorno da requisição de status da construção da GRN via API. Fonte: o autor.	35
Figura 15 – Exemplo de requisição da GRN construída via API. Fonte: o autor.	35
Figura 16 – Exemplo de retorno da requisição da GRN construída via API. Fonte: o autor. .	36
Figura 17 – Exemplo de requisição de informações de um nó da GRN via API. Fonte: o autor.	36
Figura 18 – Exemplo do retorno da requisição de informações de um nó da GRN via API. Fonte: o autor.	37
Figura 19 – Exemplo da estrutura de nós do Cytoscape.js. Fonte: o autor.	37
Figura 20 – Exemplo da estrutura de nós do Cytoscape.js customizada. Fonte: o autor. .	38
Figura 21 – Exemplo de parâmetros de configuração do Cytoscape.js. Fonte: o autor. . .	39
Figura 22 – Exemplo de parâmetros de configuração do Cytoscape.js. Fonte: o autor. . .	40
Figura 23 – Caixa de seleção do fungo para construção da GRN. Fonte: o autor.	44

Figura 24 – Visualização de detalhes do fungo selecionado para construção da GRN. Fonte: o autor.	45
Figura 25 – Visão geral da tela de seleção do fungo para construção da GRN. Fonte: o autor.	46
Figura 26 – Tela de acompanhamento do status de construção da GRN. Fonte: o autor. .	47
Figura 27 – Visualização da GRN construída para o <i>Madurella mycetomatis</i> GCA_001275765.2. Fonte: o autor.	48
Figura 28 – Filtro do TF MMYC01_200141, presente no fungo <i>Madurella mycetomatis</i> GCA_001275765.2, e exibição de suas interações. Em verde as interações re- gulatórias positivas, em vermelho as negativas e em cinza as não identificadas. Fonte: o autor.	49
Figura 29 – Exemplo da imagem obtida através do botão <i>Export View</i> , exibindo o fil- tro do TF MMYC01_200141, presente no fungo <i>Madurella mycetomatis</i> GCA_001275765.2. Fonte: o autor.	49
Figura 30 – Visualização das estatísticas gerais e das informações do nó CCM_00007, presente no fungo <i>Cordyceps militaris</i> GCA_000225605-1. Fonte: o autor. .	50
Figura 31 – Visualização da GRN construída para o <i>Madurella mycetomatis</i> GCA_001275765.2 ampliada. Fonte: o autor.	51
Figura 32 – Visualização da GRN construída para o <i>Cordyceps militaris</i> (GCA_000225605.1). Fonte: o autor.	51

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Justificativas e Contribuições	11
1.2	Objetivos	11
1.2.1	Objetivo Geral	11
1.2.2	Objetivos Específicos	11
2	REFERENCIAL TEÓRICO	12
2.1	Referencial Teórico da Área de Aplicação	12
2.1.1	Rede Regulatória de Genes	12
2.1.1.1	Ortologia	14
2.1.1.2	Sítios de Ligação de Fatores de Transcrição	15
2.1.2	Fungos	15
2.2	Referencial Computacional	16
2.2.1	<i>Frameworks Web Para Visualização de Conteúdo e Manipulação de GRNs</i> .	16
2.2.1.1	Next.JS	17
2.2.1.2	Cytoscape.js	17
2.2.1.3	Tailwind CSS	19
2.3	Trabalhos Relacionados	20
3	METODOLOGIA DE PESQUISA	24
3.1	Quanto à natureza: pesquisa aplicada	24
3.2	Quanto aos objetivos: pesquisa exploratória	24
3.3	Quanto aos procedimentos: <i>Design Science Research</i>	24
4	METODOLOGIA DE DESENVOLVIMENTO	27
4.1	Levantamento de Requisitos	28
4.1.1	Requisitos Funcionais e Requisitos não Funcionais	28
4.1.2	Casos de Uso	29
5	RESULTADOS	30
5.1	Desenhando a jornada do usuário	30
5.2	Implementando a arquitetura	31

5.3	Obtendo os dados via API	33
5.3.1	Requisição dos organismos disponíveis para construção no sistema	33
5.3.2	Envio da solicitação de construção da GRN	33
5.3.3	Consulta de status da construção da GRN	34
5.3.4	Retorno da GRN ao final do processo de construção	35
5.3.5	Consulta de informações do nó selecionado pelo usuário	36
5.4	Construindo o grafo com Cytoscape.js	37
5.5	Tornando o carregamento e visualização da GRN eficientes em termos de tempo e memória	41
6	ANÁLISE DOS RESULTADOS	43
7	CONSIDERAÇÕES FINAIS	55
	REFERÊNCIAS	56
	APÊNDICES	60
	APÊNDICE A – Documento de Requisitos	61
	APÊNDICE B – Documento de Casos de Uso	85

1 INTRODUÇÃO

Nos últimos anos, a bioinformática, área em que a computação é utilizada para compreender a biologia molecular, vem ganhando relevância (8). Com a quantidade cada vez maior de sequenciamento de DNA e RNA, se faz necessário desenvolver ferramentas que sejam capazes de processá-los. O sequenciamento do DNA do ser vivo a ser estudado pode ser do DNA completo ou de um gene de interesse do pesquisador (9).

Um dos maiores desafios da biologia molecular é identificar as relações regulatórias entre genes reguladores da transcrição e seus alvos, ou seja, os genes regulados, pois a partir disso é possível compreender os fenômenos biológicos como, por exemplo, o crescimento e divisão celular. Por conta disso, surgiu a necessidade de organizar os dados na forma de uma Rede Regulatória de Genes, do inglês *Gene Regulatory Network* (GRN) (10), que é um grafo que apresenta os reguladores gênicos, do inglês *Transcription Factors* (TF), e os genes-alvo, do inglês *Target Genes* (TG), interconectados a partir de arestas.

Um gene é expresso quando ele é transcrito. Transcrição é o processo de cópia de RNA da sequência de DNA de um gene (11), enquanto que a regulação de um gene é o processo usado para controlar o momento, localização e a quantidade em que os genes são expressos (12).

Para inferir uma GRN, podem ser utilizados recursos que são divididos em seis classes, de acordo com as abordagens empregadas: Co-expressão, Motivos em Sequência, Imunoprecipitação de Cromatina (ChIP-Seq), Ortologia, Literatura e Interação Proteína-Proteína. Quanto mais informações são agregadas, mais precisa se torna uma relação TF-TG (13).

A construção, a visualização e a manipulação de GRNs são tarefas complexas que demandam proficiência em bioinformática, pois exigem a instalação e execução de diversos softwares e a manipulação de diversos arquivos intermediários, conforme visto nas ferramentas descritas em (13). Sendo assim, a pergunta que motiva este trabalho é: como visualizar e manipular GRNs de fungos de maneira fácil, sem a instalação de softwares, em tempo hábil e consumindo pouca memória?

1.1 JUSTIFICATIVAS E CONTRIBUIÇÕES

Através das GRNs é possível entender qual é a influência de cada interação entre genes em um determinado organismo (13). Com esses dados, os pesquisadores podem obter avanços nos estudos científicos com um menor custo financeiro, já que o sequenciamento genético é um processo de alto custo.

A inferência de GRNs provou ser uma ferramenta poderosa no estudo de processos essenciais que ocorrem em sistemas celulares, como desenvolvimento, diferenciação, metabolismo, adaptação e sinalização. A aplicação de GRNs podem trazer inovação para campos de pesquisa como saúde humana e ciência de plantas (13).

As ferramentas atuais geralmente são de linha de comando, requerem muito processamento e exigem bastante conhecimento em informática para configurar e utilizar. Além disso, essas ferramentas não são voltadas à análise de GRNs de fungos, como visto em (14).

Um sistema web, que seja de fácil uso, pode facilitar o progresso de pesquisas na área, já que não é necessário ter conhecimento técnico nem poder de processamento no computador do cliente para construir e visualizar GRNs.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Desenvolver uma solução de software para a visualização e manipulação de uma GRN. Essa solução contempla o *front-end* de uma aplicação Web, sendo responsável por permitir a visualização e a configuração das características da GRN de acordo com as necessidades do usuário.

1.2.2 Objetivos Específicos

Abaixo, são listados os objetivos específicos para este trabalho:

- Projetar a jornada do usuário durante a utilização do sistema.
- Arquitetar o sistema utilizando um *framework* para visualização de redes.
- Definir uma estratégia de interface para realizar a visualização e manipulação da GRN, de maneira que seu carregamento seja eficiente em termos de tempo e memória.

2 REFERENCIAL TEÓRICO

Para a execução deste trabalho, foi realizada uma pesquisa para entendimento do estado da arte da área de estudo. Neste capítulo, serão apresentados conceitos da área de aplicação e tecnologias que serão utilizadas para atingir os objetivos propostos.

2.1 REFERENCIAL TEÓRICO DA ÁREA DE APLICAÇÃO

O entendimento de conceitos da biologia e da bioinformática são fundamentais para a construção da ferramenta proposta. Nas próximas seções, serão apresentadas algumas definições de termos da área.

2.1.1 Rede Regulatória de Genes

Compreender as relações entre genes e os produtos que eles codificam é um dos principais desafios da biologia experimental e computacional. Identificar as relações regulatórias entre os reguladores da transcrição e seus alvos é fundamental para compreender os fenômenos biológicos que variam do crescimento e divisão celular à diferenciação e desenvolvimento celular (15).

Para conseguir identificar essas relações, surgiu a necessidade de organizar os dados em forma de Rede Regulatória de Genes (GRN, do inglês *Gene Regulatory Network*), a qual consiste em um grafo direcionado em que os nós dos reguladores da expressão gênica se conectam aos nós dos genes-alvo. Os reguladores da expressão gênica incluem os fatores de transcrição (TFs, do inglês *transcription factors*), que podem estimular a ativação ou repressão dos genes-alvo (TGs, do inglês *target genes*) (1). A figura 1 apresenta um exemplo de GRN.

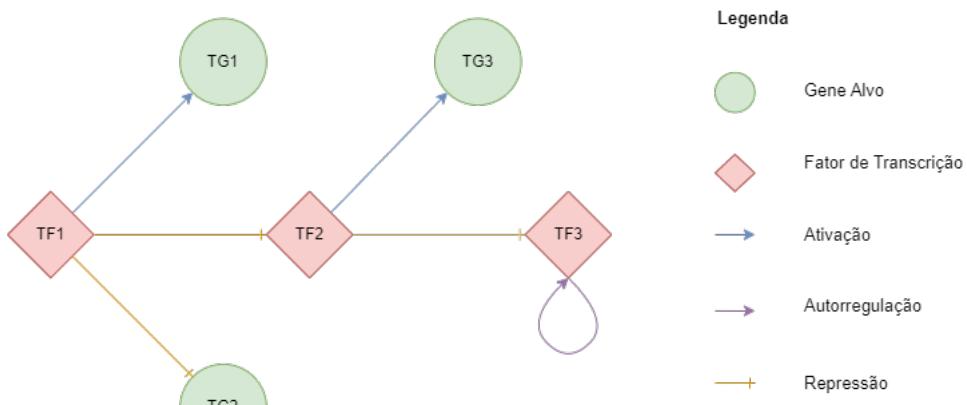


Figura 1 – Exemplo de uma GRN. Fonte: o autor.

Para realizar a construção de uma GRN, podem ser utilizados diversos recursos, que são classificados em seis tipos: Co-expressão Gênica, Motivos em Sequência, Ortologia, Imunoprecipitação de Cromatina (ChIP-Seq), Literatura e Interação Proteína-Proteína (13). A figura 2 ilustra cada uma das classes.



Figura 2 – Recursos que podem ser utilizados para inferir uma GRN. Fonte: (1)

Para a execução deste trabalho, serão visualizadas e manipuladas GRNs construídas a partir de duas fontes de dados: Ortologia e Motivos em Sequências (ou Sítios de Ligação de Fatores de Transcrição, do inglês *Transcription Factor Binding Sites* (TFBSs)).

2.1.1.1 Ortologia

Ortologia é uma relação evolutiva entre um par (ou múltiplas formas) de genes que compartilham um ancestral comum. Essa forma de relacionamento contempla similaridade de sequência, ou pelo menos, a conservação de padrões (16). Na inferência de redes genéticas, a ortologia é importante para maximizar o conteúdo e a precisão da informação. Identificar a presença de genes em diferentes espécies que podem ser rastreados até um ancestral comum e assumir que a ortologia não é apenas baseada em sequência, mas também funcional, permite a hipótese de uma rede TF-TG complexa (13). A presença de múltiplos candidatos a ortólogos de ambos os TFs e seus TGs enriquece a visão geral sobre todas as possibilidades de interações regulatórias entre esses dois elementos (17). Além disso, a previsão de uma interação em uma espécie pode ser estendida e verificada em outras espécies. A extensão da previsão de interações regulatórias entre espécies pode ser alcançada com ferramentas de bioinformática, com o objetivo de obter conhecimento cada vez mais detalhado sobre os sistemas reguladores da transcrição (17).

A Figura 3 apresenta um exemplo de ortologia. Nela, é possível verificar que a Espécie A e Espécie B são descendentes de um mesmo ancestral. Sendo assim, ambas as espécies possuem o Gene G, que pode desempenhar funções semelhantes porém adaptadas a suas características evolutivas.

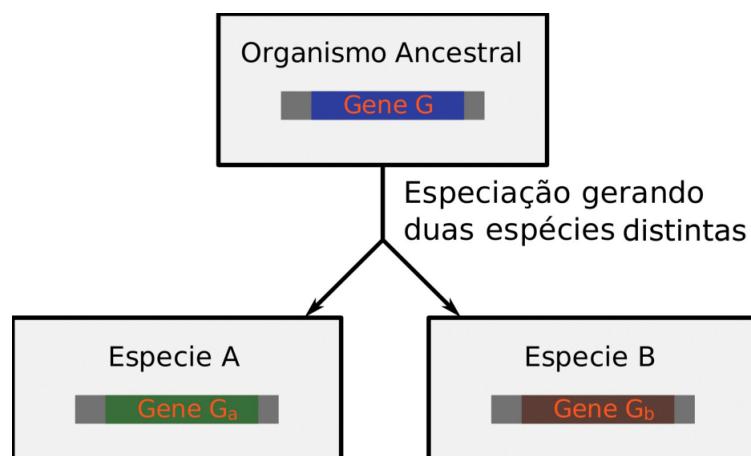


Figura 3 – Exemplo do conceito de ortologia. Fonte: (2)

2.1.1.2 Sítios de Ligação de Fatores de Transcrição

Os Sítios de Ligação de Fatores de Transcrição são áreas do DNA em que as proteínas reguladoras da expressão gênica (TFs) se ligam. A afinidade de ligação de um TF depende do seu domínio de ligação ao DNA e da sequência específica de nucleotídeos que é alvo.

Normalmente, os métodos utilizados para predição de TFBSS utilizam matrizes de peso de posição (PWMs, do inglês *position weight matrices*), também conhecidas como matrizes de pontuação específicas de posição (PSSMs, do inglês *position-specific scoring matrices*) (18). Uma matriz de peso contém valores que representam a pontuação para encontrar cada um dos quatro nucleotídeos em cada posição na sequência de DNA. Um exemplo desse processo pode ser visto na Figura 4.

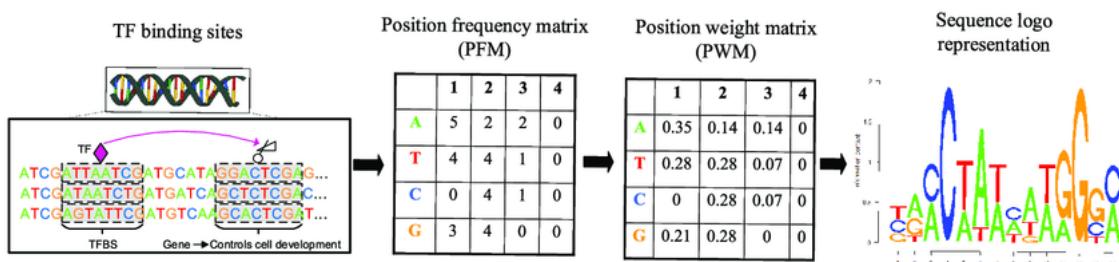


Figura 4 – Processo de construção de uma matriz de posição. Fonte: (3)

2.1.2 Fungos

Os fungos possuem diversos usos e aplicações consolidadas que são importantes para a humanidade. Mesmo assim, esses organismos continuam sendo pouco explorados devido à falta de recursos financeiros. Através dos fungos, a medicina foi beneficiada com medicamentos como a penicilina e a lovastatina, utilizados em escala global (4). Na agricultura, os fungos são utilizados para melhorar a produção de *commodities*, além de serem utilizados na produção de alimentos e bebidas. A figura 5 apresenta o processo de descoberta, estudo e potenciais áreas de aplicação dos fungos.

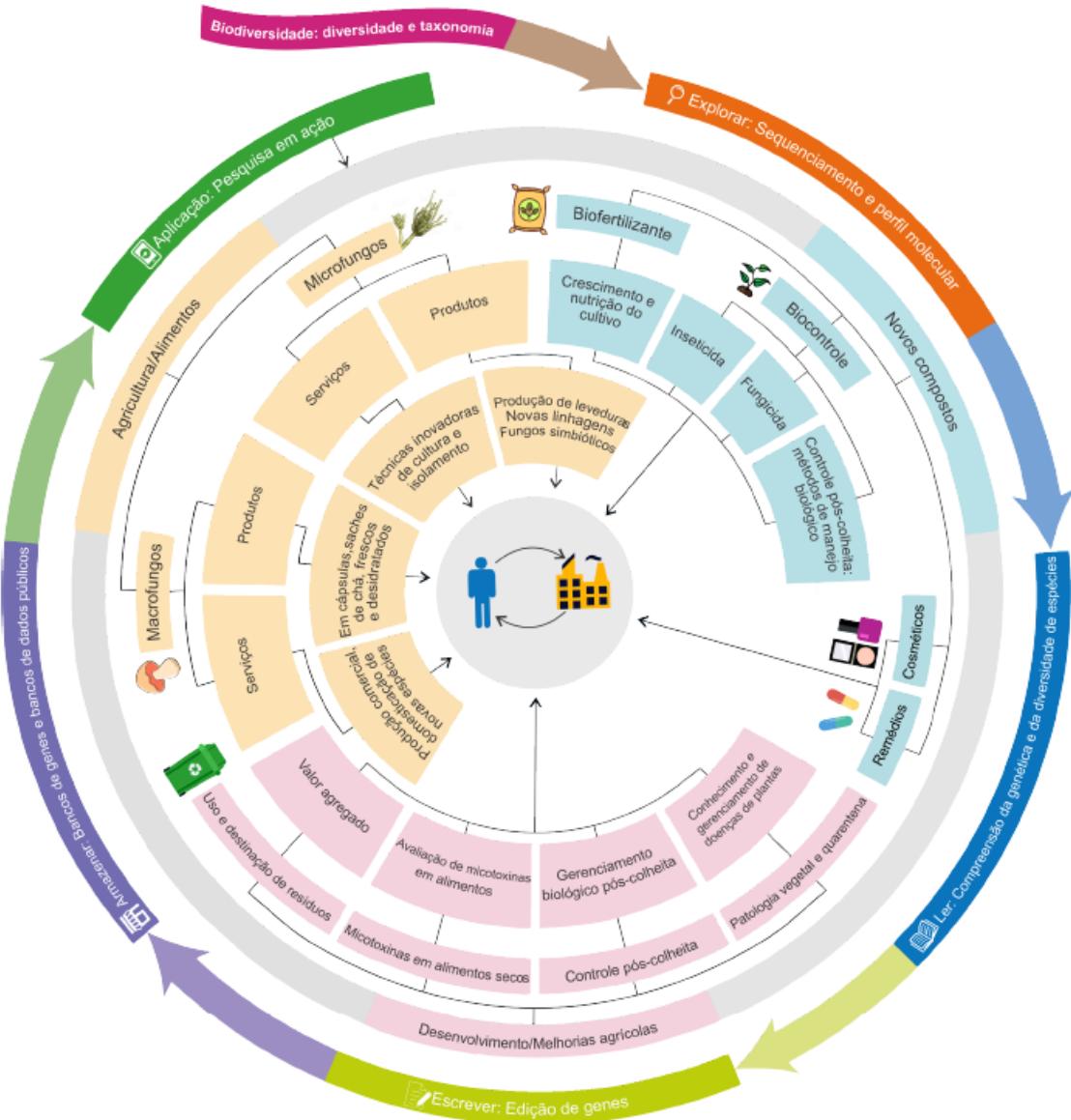


Figura 5 – Possíveis aplicações dos conhecimentos sobre fungos. Fonte: (4)

2.2 REFERENCIAL COMPUTACIONAL

A seguir, serão abordados alguns temas relacionados à aplicação de tecnologias para atingir os objetivos propostos para este trabalho.

2.2.1 Frameworks Web Para Visualização de Conteúdo e Manipulação de GRNs

Para alcançar os objetivos deste trabalho, será desenvolvida uma aplicação Web que seja capaz de visualizar e manipular GRNs. Como forma de facilitar o processo de desenvolvimento, serão utilizados alguns *frameworks* que estão descritos nas seções abaixo.

2.2.1.1 Next.JS

Next.JS é um *framework* do React, uma biblioteca da linguagem JavaScript para criação de interfaces de usuário. Através dele é possível construir aplicações Web *full-stack*, aliando recursos e otimizações adicionais com os Componentes do React (19). O Next também abstrai o processo de configuração das ferramentas necessárias para que o React funcione, permitindo que o usuário evite gastar tempo com parametrizações e foque no desenvolvimento do aplicativo (20).

Alguns dos principais recursos do Next.js são:

- **Roteamento:** um roteador baseado em sistema de arquivos construído sobre componentes de servidor que oferece suporte a layouts, roteamento aninhado, estados de carregamento, tratamento de erros, entre outros.
- **Renderização:** Renderização do lado do cliente e do lado do servidor com componentes de cliente e servidor. Otimizado ainda mais com renderização estática e dinâmica no servidor com Next.js.
- **Estilização:** Suporte para diversos métodos de estilo, incluindo Tailwind CSS.
- **Otimizações:** Otimização de imagens, fontes e scripts para melhorar as Principais Métricas da Web (21) e a experiência do usuário do seu aplicativo.

Para instalar o Next.js, é preciso ter o Node.js (22) instalado no ambiente. A maneira mais fácil de executar o processo de instalação é através da ferramenta *create-next-app*, que realiza todas as configurações necessárias para que o Next.js funcione de maneira automática (20). Após a finalização da instalação, a estrutura do projeto estará pronta para uso e poderá ser customizada para melhor atender às necessidades da aplicação (23). Sua documentação está disponível na Internet, através do seu site oficial (19).

2.2.1.2 Cytoscape.js

Cytoscape.js é um *framework* de código aberto para visualização e manipulação de grafos. Através dele, é possível integrar os grafos com anotações, perfis de expressão gênica e outros dados de estado, além de permitir interações com outras partes do sistema, com a utilização de métodos que capturam os eventos do usuário (24). Possui uma biblioteca na linguagem JavaScript que pode ser utilizada na Web e oferece suporte aos navegadores de

computadores e smartphones, incluindo também o reconhecimento de gestos com os dedos, como o de pinçar para aplicar zoom, seleção de caixa, entre outros. Também é possível utilizá-lo em modo *headless*, para fazer análise de dados em um terminal ou servidor web.

A arquitetura do *framework* é composta de dois componentes, o *core* e o *collection*. No *core* ficam as principais funcionalidades, que permitem ao programador definir *layouts*, alterar a janela de exibição e executar outras operações no grafo como um todo (25). Com ele também é possível utilizar diversas funções para acessar os elementos no grafo. Cada uma delas retorna uma *collection*, um grupo de elementos no grafo. Os componentes *collection* dispõem de funções que permitem ao programador filtrar os elementos, realizar operações, coletar informações, entre outros (25).

Os elementos (nós e arestas) e parâmetros de configuração do grafo são enviados ao componente *core* no formato JSON (26). O processo de envio das informações, com formato aceito e dados necessários pode ser consultado na documentação do *framework* e também no Capítulo 5.4.

A instalação do Cytoscape.js pode ser realizada através da importação da biblioteca JavaScript disponível em um dos três Content Delivery Networks (CDNs) indicados em sua documentação, caso ele seja utilizado em uma página Web estática, ou instalado via Node Package Manager (NPM), caso seja utilizado em um projeto de sistema que utilize Node.js.

A documentação do Cytoscape.js está disponível em seu site oficial. Nela há algumas demonstrações e mais informações a respeito de utilização (27). A Figura 6 apresenta um dos exemplos disponíveis no site oficial.

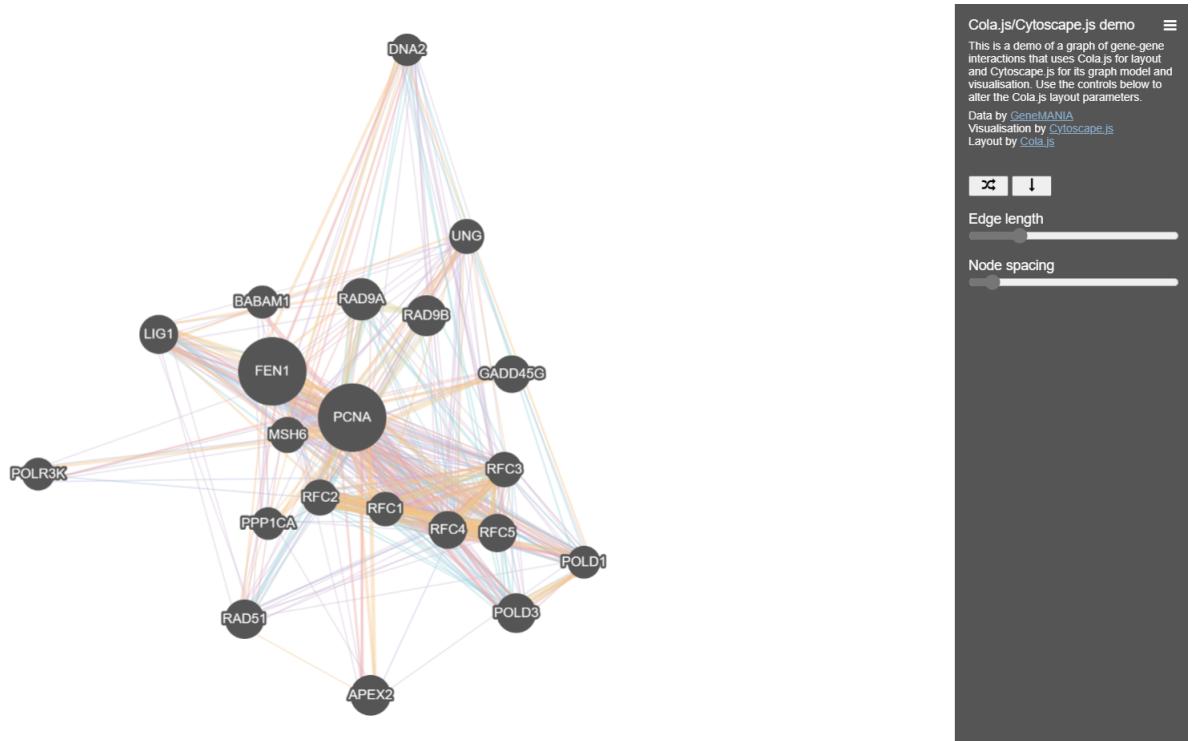


Figura 6 – Exemplo de aplicação do Cytoscape.js para visualização de grafos. Fonte: (5)

2.2.1.3 Tailwind CSS

Tailwind é um *framework Cascading Style Sheets* (CSS) (28) para estilização de páginas Web. Sua abordagem facilita a consistência na escolha dos estilos de elementos, como tipografia, cores, espaçamento, sombras e tudo mais que compõem um sistema de design bem projetado. Ele implementa classes com responsabilidade única, seguindo os fundamentos da utilidade em primeiro lugar (29). Isso significa que a classe de nome `bg-white`, por exemplo, apenas definirá a cor de fundo como branca. Essa abordagem nos permite implementar um componente completamente customizado sem escrever uma linha de CSS.

As classes de estilo disponibilizadas pelo *framework* fazem parte de um sistema de design predefinido, que também pode ser customizado pelo programador, caso necessário. Um sistema de design é uma coleção de componentes reutilizáveis, guiados por padrões claros, que podem ser montados para construir qualquer número de aplicações (30).

Algumas outras vantagens de se utilizar o Tailwind são:

- Facilidade de criar layouts responsivos utilizando pontos de quebra predefinidos.
- Possibilidade de estilizar elementos em estados de interação do usuário utilizando modificadores condicionais.

- Reutilização de estilos para facilitar a manutenção de código.
- Capacidade de customizar o *framework* para atender as necessidades do projeto.

Ele pode ser instalado de diversas maneiras, desde a partir de um CDN até através do Node.js. Sua documentação contém as classes do seu sistema de design e outras informações úteis para utilização do framework (31).

2.3 TRABALHOS RELACIONADOS

Durante as pesquisas realizadas para o desenvolvimento deste trabalho, foram identificados alguns trabalhos correlatos. A busca pelos trabalhos já publicados é fundamental para o entendimento do estado da arte, possibilitando a identificação de métodos, tecnologias e ferramentas que podem ser utilizadas para agregar valor às contribuições propostas durante a produção de novos trabalhos acadêmicos.

O *FunRegulation* (32) é uma ferramenta desenvolvida para a construção e visualização de GRNs de três fungos específicos, o *Penicillium ucsense*, *Penicillium oxalicum* e *Fusariun oxysporum*. Todo o processo de utilização desta ferramenta se dá por meio de linha de comando e necessita de intervenção do usuário durante o processo. O GRNCOP2 (33) é uma ferramenta utilizada para estudo da doença de Alzheimer, que utiliza o Cytoscape Desktop para a visualização de GRNs. O Pheno2GRN (34) é uma ferramenta para identificação de fenótipos, através de uma ferramenta acessível pelo navegador, que utiliza Java Web e Cytoscape.js. Por fim, o PTHGRN (35) é uma ferramenta Web para visualização de GRNs de humanos, camundongos, ratos, *Drosophila melanogaster* e *Caenorhabditis elegans*. Ela é uma ferramenta Web e foi construída utilizando PHP, Pascal e Cytoscape.js.

Na Tabela 0 abaixo, são descritos os principais trabalhos encontrados durante as pesquisas para o desenvolvimento deste trabalho.

Tabela 0 – Principais trabalhos relacionados.

Ferramenta	Alvo(s)	Ambiente	Tecnologias	DOI	Descrição
FunRegulation (32)	<i>Penicillium ucsense</i> , <i>Penicillium oxalicum</i> e <i>Fusarium oxysporum</i>	Linha de comando	Python, Bi- oPython e SQLite	10.3389/ fmicb.2020. 588263	Construção da GRN para análise de suas interações regulatórias. Etapas do processo são realizadas manualmente.
GRNCOP2 (33)	Doença de Alzheimer	Desktop	Cytoscape Desktop	10.1016/ j.cmpb.2019. 05.030	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 como exemplo.

Continua na próxima página

Tabela 0 – continuação da página anterior

Ferramenta	Alvo(s)	Ambiente	Tecnologias	DOI	Descrição
Pheno2GRN (34)	Fenótipos	Web	Java e Cytoscape.js	10.1145/2649387.2660851	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.
PTHGRN (35)	Humano, camundongo, rato, <i>Drosophila melanogaster</i> e <i>Cae-norhabditis elegans</i> .	Web	PHP, Pascal e Cytoscape.js	10.1093/nar/gku471	Apresenta 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.

Das ferramentas encontradas, apenas a *FunRegulation* pode ser utilizada para vi-

sualizar GRNs de três fungos específicos. Nenhuma das outras ferramentas apresentadas nas publicações encontradas podem ser utilizadas para visualizar e manipular GRNs de fungos. Também em nenhuma delas é possível executar todo o processo de construção, visualização e manipulação de maneira automatizada. Em algum momento da execução, é preciso que haja intervenção do pesquisador para chegar à etapa de visualização da rede.

3 METODOLOGIA DE PESQUISA

Neste capítulo será abordada a modalidade de pesquisa adequada ao objeto de pesquisa, identificando o seu tipo de pesquisa em relação à sua natureza, seus objetivos e seus procedimentos.

3.1 QUANTO À NATUREZA: PESQUISA APLICADA

A pesquisa aplicada tem como objetivo produzir conhecimento para aplicação prática, dirigindo-o a solução de problemas específicos. Em relação a esta pesquisa, é observada a falta de uma ferramenta web que seja simples de se utilizar e que produza GRNs para fungos. O objeto de pesquisa se encaixa na pesquisa aplicada, pois o objetivo deste trabalho é produzir uma ferramenta que preencha essa lacuna.

3.2 QUANTO AOS OBJETIVOS: PESQUISA EXPLORATÓRIA

A pesquisa exploratória tem como objetivo proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito ou a construir hipóteses. A maioria dessas pesquisas envolve: (a) levantamento bibliográfico; (b) consultas com especialistas que tiveram experiências práticas com o problema pesquisado; e (c) análise de exemplos que estimulem a compreensão. Pesquisas exploratórias geralmente sugerem procedimentos de pesquisa bibliográfica e estudo de caso (36). Este trabalho se encaixa na pesquisa exploratória, pois será necessário realizar uma pesquisa sobre o estado da arte e adaptar os recursos encontrados para atingir os objetivos definidos.

3.3 QUANTO AOS PROCEDIMENTOS: *DESIGN SCIENCE RESEARCH*

O DSR é um modelo de pesquisa que tem dois objetivos: desenvolver um artefato para resolver um problema prático num contexto específico e gerar novos conhecimentos técnicos e científicos (6). Neste projeto, será aplicado o *Design Science Research Model*, uma abordagem do DSR que visa projetar um artefato baseado em conjecturas comportamentais para resolver um problema. Para validar a solução do problema através da utilização do artefato

projetado, é realizada uma avaliação empírica que verifica se as conjecturas comportamentais que fundamentaram a criação do artefato são aceitáveis. Seguindo esses passos, são produzidos conhecimentos técnicos e científicos sobre o problema abordado (6). A Figura 7 descreve a aplicação do DSR neste projeto.

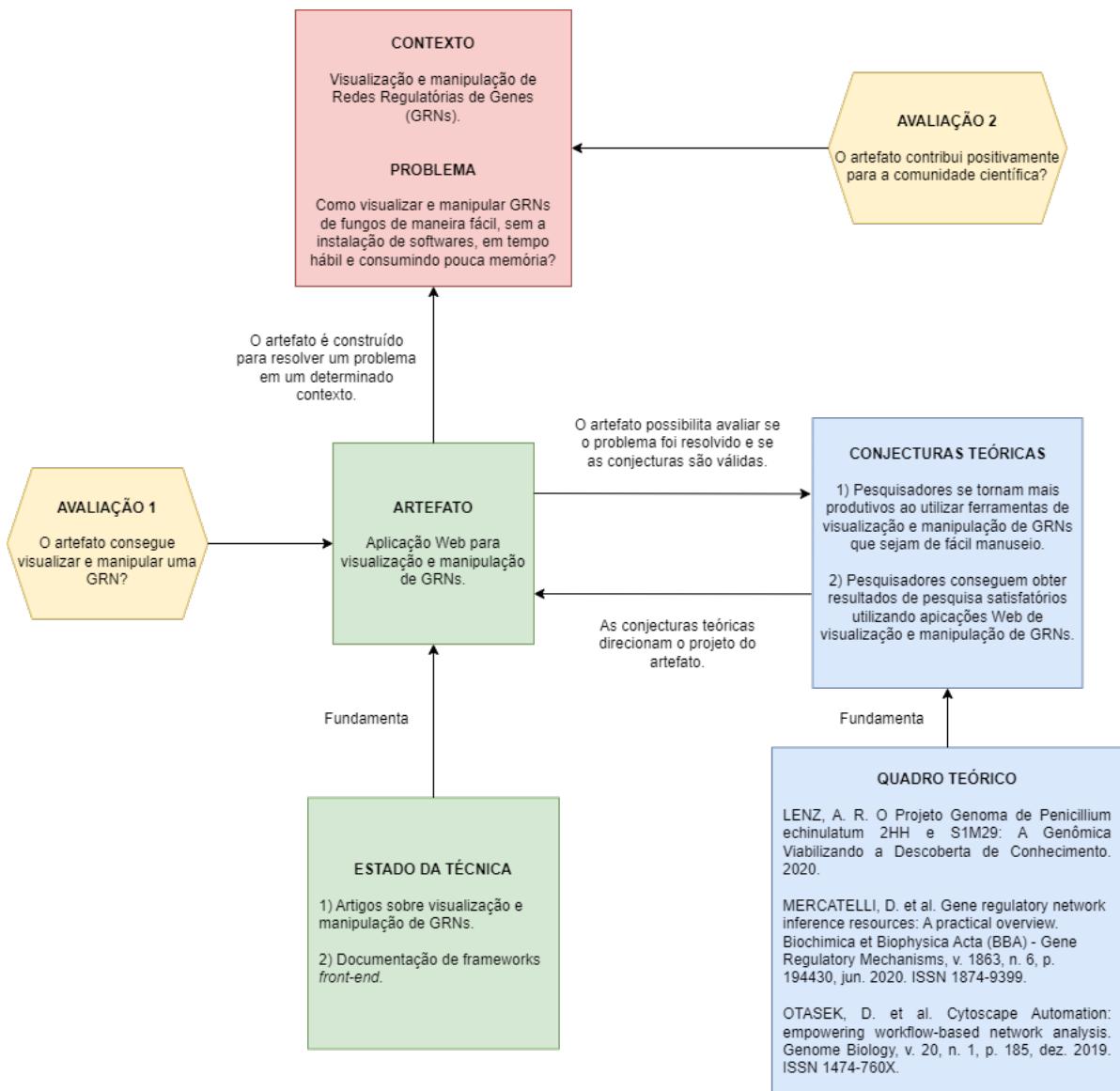


Figura 7 – Modelo *Design Science Research* para o projeto. Fonte: o autor, baseado em Pimentel (2020) (6).

Para realizar a avaliação 1 do artefato, exibida na Figura 7, que questiona se o artefato consegue visualizar e manipular uma GRN, foram definidos alguns passos a serem seguidos por um pesquisador da área. Para validar a capacidade da ferramenta de cumprir com seu objetivo, foi estipulado que ela deve ser capaz de solicitar a construção ao back-end, visualizar e ser capaz de manipular duas GRNs de fungos de ordens distintas.

O fluxo da avaliação segue as etapas definidas na jornada do usuário, apresentada no Capítulo 5.1, e contempla a seleção do fungo, configuração de parâmetros, solicitação de construção da GRN, acompanhamento do processo de construção, visualização e manipulação da GRN através da interação com o grafo, alteração de zoom, layout, filtro de TFs e exportação da visualização.

Para realizar a avaliação 2, que questiona se o artefato contribui positivamente para a comunidade científica, foi definido que o sistema precisa ser disponibilizado abertamente na Web, para que a comunidade científica seja capaz de acessar e testar a ferramenta. Após esta etapa, é necessária a realização de uma entrevista/questionário para concluirmos se o artefato realmente contribui positivamente e identificarmos possíveis pontos de melhoria. A disponibilização da ferramenta na Internet e a avaliação da contribuição para a comunidade científica serão trabalhos futuros.

4 METODOLOGIA DE DESENVOLVIMENTO

Neste capítulo, serão abordadas as ferramentas e metodologias para desenvolvimento do *front-end* do FunRegulation.

Para definir o escopo do sistema, foram construídos os documentos de definição de requisitos e análise de requisitos. O documento de definição de requisitos aborda os requisitos funcionais, que são recursos que o sistema deve ser capaz de fazer, e os requisitos não funcionais, que se referem a características do sistema, como desempenho, usabilidade e segurança. O Anexo A contém todos os requisitos definidos, enquanto que o Capítulo 4.1.1 contém os principais requisitos que se referem ao *front-end* da ferramenta.

O documento de análise de requisitos, disponibilizado por completo no Anexo B, contém o diagrama de casos de uso e a descrição de casos de uso. A descrição de casos de uso descreve como o usuário interage com o sistema para utilizar as funcionalidades disponíveis. Por conta da limitação de tempo para o desenvolvimento do trabalho, foram selecionados os casos de uso considerados essenciais para o funcionamento do sistema. Eles são listados no Capítulo 4.1.2.

O Scrum, metodologia de desenvolvimento ágil utilizada no desenvolvimento de *softwares* (37), foi a metodologia escolhida para o projeto. A razão da escolha está na flexibilidade de divisão, entrega e revisão das tarefas, permitindo que ocorram pequenas entregas durante a execução do projeto, denominadas de *Sprint* (7). Para que ela fosse melhor encaixada ao projeto, definiu-se que a *Sprint* teria duração de sete dias e as reuniões para alinhamento do projeto ocorreriam uma vez por semana, conforme demonstrado na Figura 8.

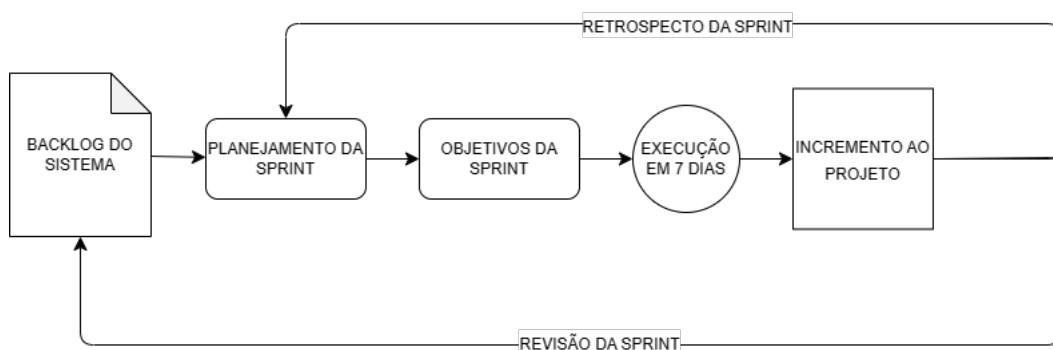


Figura 8 – Metodologia Scrum adaptada para o FunRegulation. Fonte: o autor, adaptada de (7)

Juntamente com o Scrum, foi utilizado o Kanban para acompanhar o andamento das tarefas selecionadas para cada *Sprint*. O método Kanban no desenvolvimento de software leva as equipes de projeto a visualizar o fluxo de trabalho, limitar o trabalho em andamento (WIP, do inglês *Work in Progress*) em cada estágio do fluxo de trabalho e medir o tempo do ciclo (38). O quadro Kanban dá visibilidade ao processo de software, comunica claramente as prioridades e destaca os gargalos. Além disso, seu objetivo é minimizar o WIP, ou seja, desenvolver apenas os itens solicitados (39). Para aplicar o método Kanban foi utilizado o Trello, ferramenta online para gestão de quadros e tarefas (40).

Para controlar o versionamento do sistema foi utilizado o GIT, através do GitHub, a ferramenta de versionamento com a maior comunidade da atualidade (41). O repositório contendo o código-fonte do *front-end* do FunRegulation está disponível no GitHub (42).

4.1 LEVANTAMENTO DE REQUISITOS

Os requisitos do sistema foram definidos com base nos objetivos propostos para este projeto. A seguir, serão listados os requisitos funcionais e casos de uso para o *front-end* da aplicação.

4.1.1 Requisitos Funcionais e Requisitos não Funcionais

Dentre os requisitos funcionais propostos para este projeto, os que fazem parte do *front-end* da aplicação são:

- RF09 - Exibir e Manipular GRN
- RF10 - Exibir Anotação Funcional de um Nô
- RF11 - Exibir Estatísticas
- RF12 - Configurar GRN
- RF13 - Permitir *Zoom*
- RF14 - Escolher Layout de Exibição
- RF15 - Filtrar Interações Regulatórias
- RF16 - Manter GRNs e *Views*
- RF17 - Exportar GRN ou *View*
- RF18 - Mostrar Tutorial
- RF19 - Mostrar Tela Inicial

Os requisitos não funcionais referentes ao *front-end* do *FunRegulation* são:

- RN05 - *Upload* de arquivos
- RN06 - Identificar elementos da GRN
- RN07 - Exibir arestas em cores distintas
- RN08 – Salvar configurações automaticamente

Os requisitos não são listados a partir do número 1 pois apenas os relacionados ao *front-end* estão listados. Todos os requisitos funcionais, não funcionais e seus detalhes podem ser consultados no Apêndice A.

4.1.2 Casos de Uso

Os casos de uso voltados ao front-end da aplicação estão a seguir:

- UC06 - Construir GRN
- UC09 - Exibir e Manipular GRN
- UC10 - Exibir Anotação Funcional
- UC11 - Exibir Estatísticas
- UC12 - Configurar GRN
- UC13 - Permitir *Zoom In/Out*
- UC14 - Escolher Layout
- UC15 - Filtrar Interações Regulatórias
- UC16 - Filtrar TFs
- UC17 - Manter GRNs e *Views*
- UC18 - Exportar GRN ou *View*
- UC19 - Exibir Telas Informativas

Os casos de uso não são listados a partir do número 1 ou podem estar fora de ordem pois apenas os relacionados ao *front-end* estão listados. Todos os casos de uso e seus detalhes podem ser consultados no Apêndice B.

5 RESULTADOS

Neste capítulo, serão abordadas as etapas de desenvolvimento do sistema, desde o desenho da jornada do usuário, realização das requisições ao *back-end* via API e codificação das telas de construção e visualização das GRNs.

5.1 DESENHANDO A JORNADA DO USUÁRIO

Para que o *FunRegulation* fosse de fácil uso e cumprisse com sua proposta, se fez necessário construir a jornada do usuário de maneira a identificar os passos do especialista da área dentro do sistema, de modo a tornar as funcionalidades simples, acessíveis e fáceis de identificar. Deste modo, foi desenhada a jornada para navegação do usuário no sistema, contemplando as ações de seleção do fungo, construção da rede, visualização e manipulação da GRN e filtro dos TFs de interesse. A Figura 9 abaixo, apresenta os passos realizados pelo usuário durante a utilização do sistema.

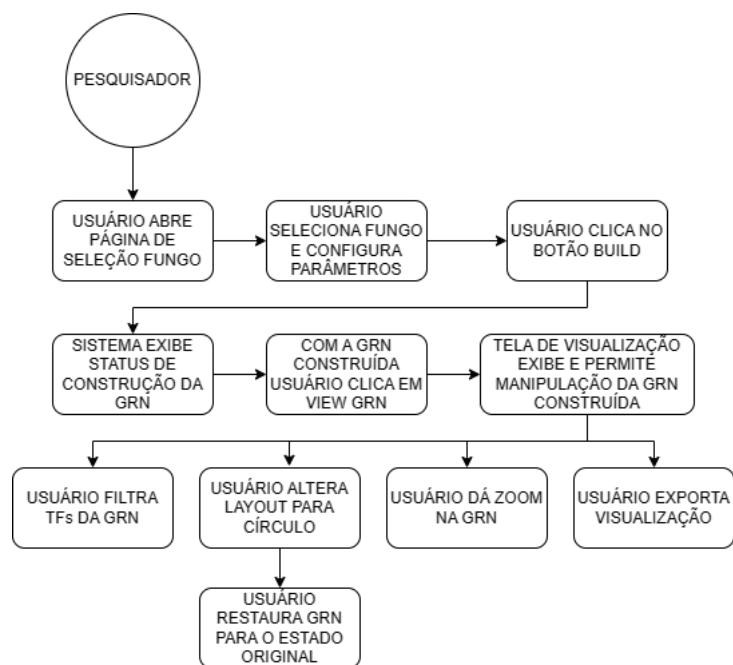


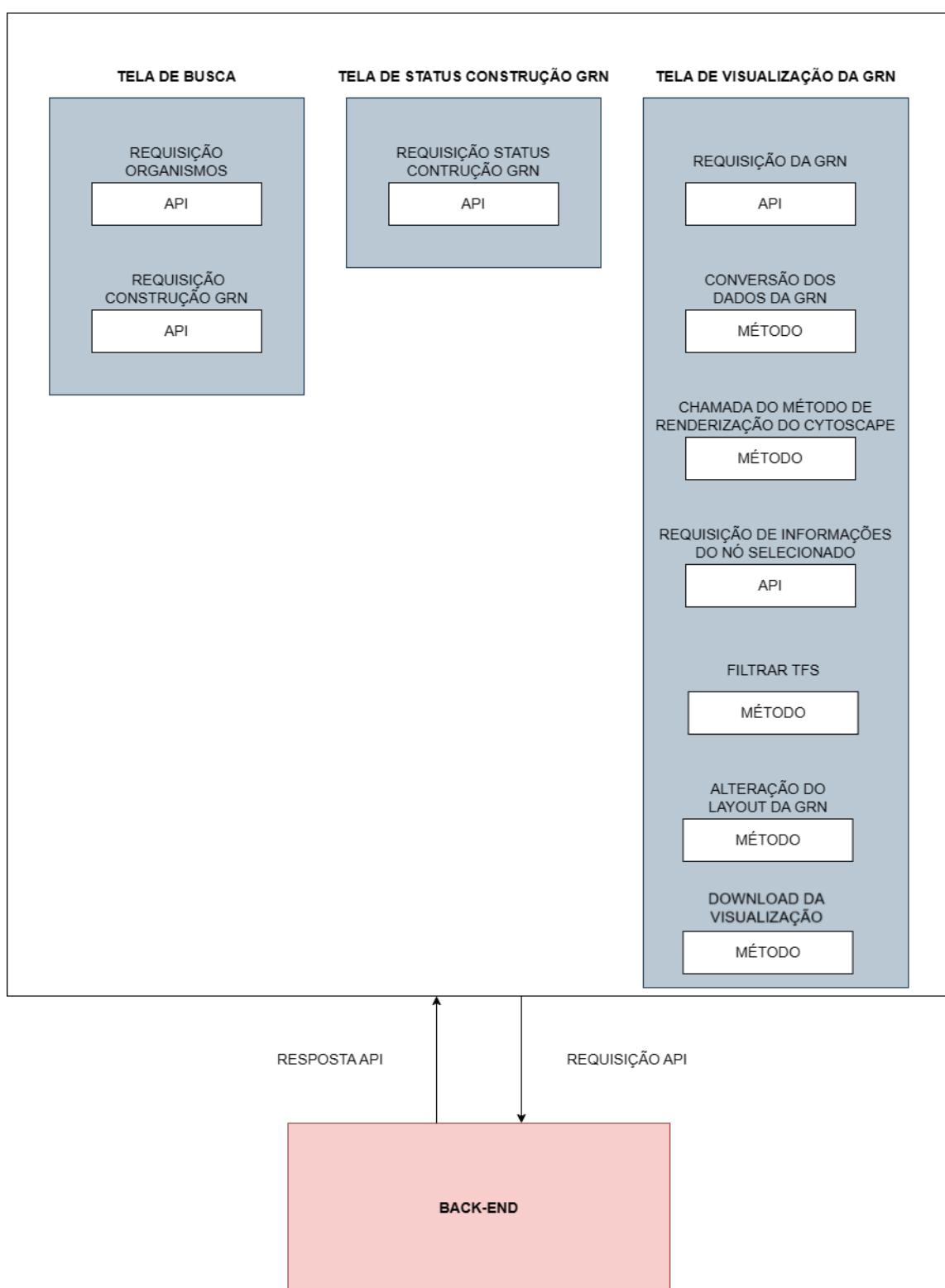
Figura 9 – Jornada do usuário para o *FunRegulation*. Fonte: o autor.

5.2 IMPLEMENTANDO A ARQUITETURA

A arquitetura do *front-end* do FunRegulation contém componentes reutilizáveis, que provém do React, base do Next.js. Dessa forma é possível aproveitar um mesmo componente em várias partes do sistema.

Cada uma das telas possui seus próprios métodos e em alguns casos eles realizam requisições para o *back-end* do FunRegulation. A Figura 10 apresenta a arquitetura do sistema.

NEXT.JS + CYTOSCAPE.JS

Figura 10 – Arquitetura do *front-end* do FunRegulation. Fonte: o autor.

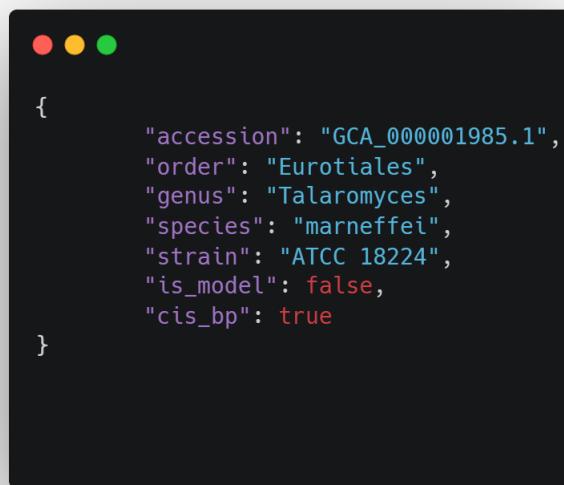
No próximo capítulo, será descrita a forma de funcionamento de cada um dos *endpoints* da API, utilizados nas páginas do sistema.

5.3 OBTENDO OS DADOS VIA API

Para visualizar as GRNs no *front-end*, era necessário receber os dados vindos do *back-end* do FunRegulation. Por conta disso, foram desenvolvidos métodos de requisição à API disponibilizada por ele, que responde com os dados em formato JSON. Nas seções seguintes, serão listados os *endpoints* da API e suas respectivas funcionalidades.

5.3.1 Requisição dos organismos disponíveis para construção no sistema

Para listar os organismos disponíveis na página de busca, é necessário realizar esta requisição para a API do FunRegulation. Nesta requisição, que é do tipo GET, não se faz necessário enviar nenhum *payload*. Um exemplo do retorno obtido está contido na Figura 11.

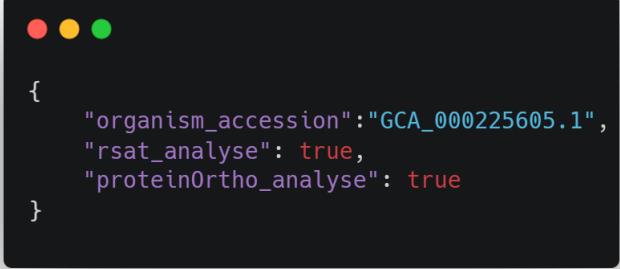


```
{
    "accession": "GCA_000001985.1",
    "order": "Eurotiales",
    "genus": "Talaromyces",
    "species": "marneffei",
    "strain": "ATCC 18224",
    "is_model": false,
    "cis_bp": true
}
```

Figura 11 – Exemplo de retorno da requisição de organismos via API. Fonte: o autor.

5.3.2 Envio da solicitação de construção da GRN

Após selecionar um dos organismos e escolher a forma como a rede será construída na tela de busca, o usuário pode enviar a solicitação ao sistema. Ao clicar no botão ”Buid”, esta requisição é enviada, contendo o *payload* presente na Figura 12.



```
{
  "organism_accession": "GCA_000225605.1",
  "rsat_analyse": true,
  "proteinOrtho_analyse": true
}
```

Figura 12 – Exemplo de requisição de construção da GRN via API. Fonte: o autor.

Ao completar o envio, o *front-end* recebe um status. Caso retorne com o status HTTP 200, que se refere a um envio bem-sucedido, é recebido também um ID de trabalho, que poderá ser utilizado para consultar o status de construção da GRN e posteriormente solicitar a rede quando o processo for finalizado. Na Figura 13 está um exemplo do retorno da requisição, em caso de sucesso.



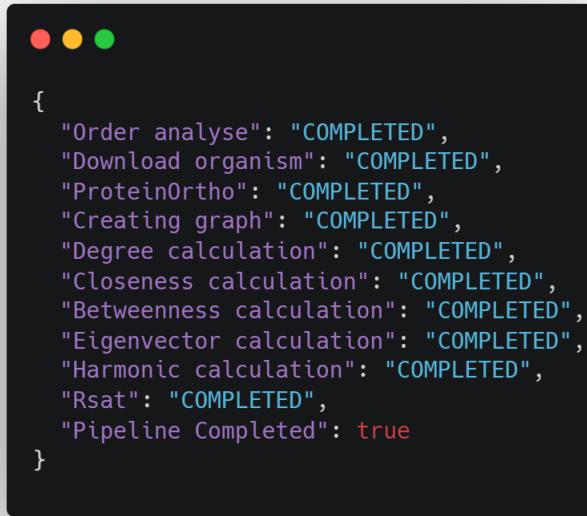
```
{
  "job_id": "0001"
}
```

Figura 13 – Exemplo de retorno da requisição de construção da GRN via API. Fonte: o autor.

5.3.3 Consulta de status da construção da GRN

Para verificar em qual estágio de construção a GRN solicitação se encontra, é realizada uma requisição que retorna o estado de cada uma das tarefas realizadas pelo *back-end*. O *payload* enviado na requisição é semelhante ao retorno da requisição de construção da GRN, na Figura 13.

Após envio da solicitação, caso retorne com o status HTTP 200, é recebido o status de cada uma das tarefas executadas, conforme Figura 14.



```
{
  "Order analyse": "COMPLETED",
  "Download organism": "COMPLETED",
  "ProteinOrtho": "COMPLETED",
  "Creating graph": "COMPLETED",
  "Degree calculation": "COMPLETED",
  "Closeness calculation": "COMPLETED",
  "Betweenness calculation": "COMPLETED",
  "Eigenvector calculation": "COMPLETED",
  "Harmonic calculation": "COMPLETED",
  "Rsat": "COMPLETED",
  "Pipeline Completed": true
}
```

Figura 14 – Exemplo de retorno da requisição de status da construção da GRN via API. Fonte: o autor.

5.3.4 Retorno da GRN ao final do processo de construção

Após a conclusão do processo de construção da GRN, o *front-end* solicita seus TFs, TGs e suas respectivas arestas para em seguida realizar a construção do grafo que será exibido em tela. Para isso, é realizada uma requisição contendo as seguintes informações presentes na Figura 15.



```
{
  "job_id": "0001"
}
```

Figura 15 – Exemplo de requisição da GRN construída via API. Fonte: o autor.

Caso a requisição seja retornada com status HTTP 200 (43), as informações são recebidas no formato presente na Figura 16.



```
{
  "tfs": [{"tf_locus_tag": "TF1"}, {"tf_locus_tag": "TF2"}],
  "tgs": [{"tg_locus_tag": "TG1"}, {"tg_locus_tag": "TG2"}],
  "edges": [{"id": 1, "tf_locus_tag": "TF1", "tg_locus_tag": "TG1", "regulatory_function": "Not applicable", "pubmedid_source": "00000001"}, {"id": 2, "tf_locus_tag": "TF2", "tg_locus_tag": "TG2", "regulatory_function": "Negative", "pubmedid_source": "00000002"}]
}
```

Figura 16 – Exemplo de retorno da requisição da GRN construída via API. Fonte: o autor.

5.3.5 Consulta de informações do nó selecionado pelo usuário

Uma das funcionalidades disponíveis na visualização da GRN é o filtro de TFs. O pesquisador pode, caso deseje, filtrar um ou mais TFs. Desta forma, ele pode concentrar a visualização nos nas proteínas de seu interesse de estudo.

Para que o sistema exiba as informações do nó selecionado pelo usuário, é necessário realizar uma requisição para o *back-end*. O *payload* conterá apenas o nome do nó selecionado, conforme demonstrado na Figura 17. A Figura 18 apresenta o formato de retorno da requisição.



```
{
  "node_id": "TF1"
}
```

Figura 17 – Exemplo de requisição de informações de um nó da GRN via API. Fonte: o autor.

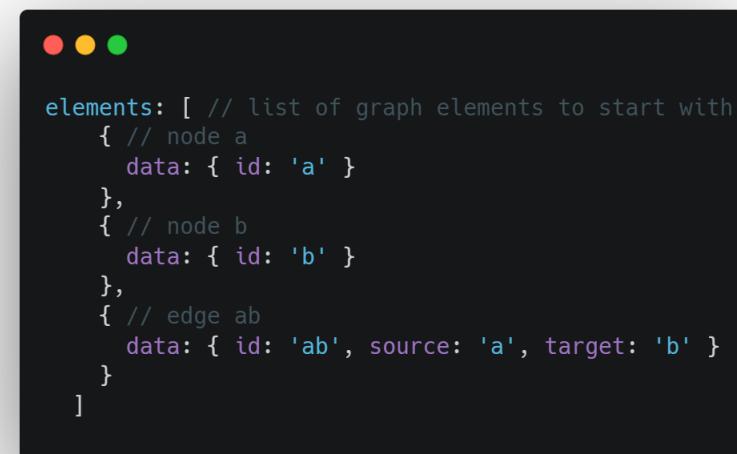


```
{
  "id": "EGX95353.1",
  "product": "37S ribosomal protein S12",
  "interpro": "IPR002594",
  "pfam": "PF01670",
  "go": "0005576",
  "gene3d": "P22669",
  "reactome": "R-HSA-114547",
  "panther": "PTHR34002",
  "uniprot": "Q14405",
  "ec_number": "3.2.1.4",
  "cazy": "GH12"
}
```

Figura 18 – Exemplo do retorno da requisição de informações de um nó da GRN via API. Fonte: o autor.

5.4 CONSTRUINDO O GRAFO COM CYTOSCAPE.JS

Para renderizar o grafo na tela com o Cytoscape.js, é preciso adequar os dados recebidos através da requisição API descrita no capítulo 5.3.4. O formato utilizado pelo *framework* está exemplificado na Figura 19.



```
elements: [ // list of graph elements to start with
  { // node a
    data: { id: 'a' }
  },
  { // node b
    data: { id: 'b' }
  },
  { // edge ab
    data: { id: 'ab', source: 'a', target: 'b' }
  }
]
```

Figura 19 – Exemplo da estrutura de nós do Cytoscape.js. Fonte: o autor.

Com base no formato aceito pelo Cytoscape.js, a lista de TFs e TGs trazidos separadamente pelo *back-end* precisa ser unificada, para que os nós sejam gerados. No campo id de cada nó, é inserido o tf_locus_tag para os TFs e o tg_locus_tag para os TGs. Além disso, para facilitar a identificação durante a visualização, é inserida uma informação adicional contendo o formato que o nó será exibido em tela, sendo ele triangle para TFs e ellipse para TGs.

As arestas devem conter o nó fonte e o alvo. Dessa forma, o id de cada aresta contém o nome do TF e TG separados por um traço, conforme o exemplo a seguir: TF1-TG1. Também pensando na facilidade de identificação da interação entre os nós, são acrescentadas duas informações adicionais, dependendo do tipo de interação que ocorra. Para interações positivas, é adicionada à variável que especifica o formato da ponta da aresta o valor triangle e para a variável que indica a cor da aresta o valor green. Para as interações negativas, são adicionados os valores tee e red, respectivamente, e para as ligações em que não é identificada nenhuma interação, os valores none e grey, respectivamente.

No final, os nós e arestas do grafo terão o formato presente na Figura 20.

```

elements: [ // list of graph elements to start with
    { // node a
        data: {id: 'a', shape: 'ellipse'}
    },
    { // node b
        data: {id: 'b', shape: 'triangle'}
    },
    { // edge ab
        data: {id: 'a-b', source: 'a', target: 'b', regulatory_function: 'triangle', regulatory_color:'green'}
    }
]
  
```

Figura 20 – Exemplo da estrutura de nós do Cytoscape.js customizada. Fonte: o autor.

Outro processo que precisa ser realizado é a parametrização do *layout* a ser utilizado na renderização da GRN. Para facilitar a visualização dos nós e suas respectivas ligações, foi escolhido o *layout* do tipo *grid* como padrão para a visualização da rede. Por padrão, o Cytoscape.cs oferece em seu código base alguns layouts predefinidos e o *layout* do tipo *grid* é um deles. Para melhorar a visibilidade da rede, foram realizadas mudanças em alguns parâmetros predefinidos pelo *framework*. Na Figura 21 estão os parâmetros alterados.



Figura 21 – Exemplo de parâmetros de configuração do Cytoscape.js. Fonte: o autor.

O que cada um dos parâmetros é responsável por fazer está listado abaixo:

- **fit**: define se o grafo renderizado respeitará o tamanho da área definida para visualização ou se ela poderá ser ultrapassada.
- **padding**: define o distanciamento da borda da área de visualização.
- **rows**: define o número de linhas que o grafo terá.
- **cols**: define o número de colunas que o grafo terá.
- **avoidOverlap**: define se os nós poderão se sobrepor.
- **spacingFactor**: define o espaço entre os nós da rede.

Por fim, para melhor identificação dos TFs e TGs, além de facilitar a identificação das ligações, foram alterados alguns parâmetros de estilo do grafo. Aqui, as informações adicionais inseridas em cada um dos nós e ligações da rede foram utilizadas para definir as mudanças desejadas.

As configurações de estilo definidas para a renderização da GRN podem ser vistas na Figura 22, que está abaixo.



```

style: [
  {
    selector: "node",
    style: {
      "background-color": "grey",
      label: "data(id)",
      shape: "data(shape)",
    },
  },
  {
    selector: "edge",
    style: {
      width: 3,
      "line-color": "data(regulatory_color)",
      "target-arrow-color": "data(regulatory_color)",
      "target-arrow-shape": "data(regulatory_function)",
      "curve-style": "bezier",
    },
  },
]

```

Figura 22 – Exemplo de parâmetros de configuração do Cytoscape.js. Fonte: o autor.

A lista dos parâmetros e suas funções está abaixo:

- **selector**: define se o estilo será aplicado aos nós ou arestas.
- **background-color**: define a cor do nó.
- **label**: define a legenda de cada nó. Aqui é utilizado o nome de cada nó, inserido anteriormente na adequação dos dados recebidos da API para o formato aceito pelo Cytoscape.
- **shape**: define o formato do nó. Aqui é utilizado o formato inserido como informação adicional de cada nó, diferenciando-o caso seja um TF ou TG.
- **width**: define a largura da linha que representa as ligações entre nós.
- **line-color**: define a cor da linha que representa a ligação entre nós. Aqui é utilizada a cor inserida anteriormente como uma das informações adicionais, levando em consideração se a interação é positiva, negativa ou não identificada.
- **target-arrow-color**: define a cor da forma exibida na ponta da linha que representa a ligação entre nós. Aqui é utilizada a mesma cor definida em line-color.

- `target-arrow-shape`: define a forma exibida na ponta da linha que representa a ligação entre nós. Aqui é utilizada a forma inserida como uma das informações adicionais de cada uma das arestas, levando em consideração se a interação é positiva, negativa ou não identificada.
- `curve-style`: define a forma de renderização das arestas.

5.5 TORNANDO O CARREGAMENTO E VISUALIZAÇÃO DA GRN EFICIENTES EM TERMOS DE TEMPO E MEMÓRIA

Para tornar o *front-end* mais leve para o usuário, foram adotadas algumas estratégias durante o seu desenvolvimento. O primeiro grupo de estratégias está relacionado à requisição e manipulação dos dados necessários para a construção da GRN. Os dados da rede são requisitados em dois momentos: no primeiro, ao solicitar as informações, a solicitação via API é respondida apenas com os nós e arestas. No segundo momento, quando o usuário clica em um determinado nó, as informações referentes àquele nó específico são requisitadas via API. Essa estratégia faz com que a quantidade de memória necessária para que o sistema funcione seja menor, já que apenas as informações complementares dos nós selecionados serão carregadas. O segundo grupo de estratégias está relacionado ao Cytoscape.cs. Em sua documentação, há uma seção voltada à otimização de performance do *framework*. Algumas das recomendações listadas nesta seção foram utilizadas, tais como a utilização do método `getElementById` nativa do Cytoscape.cs para filtrar os nós da GRN, estilização dos nós e arestas de forma nativa, utilização de cores sólidas como fundo dos elementos do grafo, utilização de formas mais simples para indicar a interação entre nós, ocultação das arestas durante interações de movimento no grafo e utilização de *cache* da visualização para tornar a renderização durante a interação menos custosa.

A seguir, essas recomendações são explicadas de maneira mais detalhada:

- **Utilização do método `getElementById` nativa do Cytoscape para filtrar os nós da GRN:** segundo sua documentação, utilizar o método `getElementByID` é a maneira mais rápida de se encontrar o nó procurado.
- **Estilização de nós e arestas de forma nativa:** A estilização dos nós e arestas foi feita seguindo a recomendação de utilizar o método nativo data ao invés de uma função própria, que tornaria esse processo mais custoso.
- **Utilização de cores sólidas como fundo dos elementos do grafo:** utilizar cores sólidas

como fundo dos elementos torna a tarefa de renderização mais simples, já que não há formas complexas a serem renderizadas.

- **Utilização de formas mais simples para indicar a interação entre nós:** apesar de precisar utilizar o método *bezier*, mais custoso do que outras opções de renderização, por conta das ligações entre o próprio nó, as formas escolhidas para indicar o tipo de interação entre os nós têm linhas retas, tornando a tarefa de renderização menos custosa.
- **Ocultação das arestas durante interações de movimento no grafo:** ocultar das arestas durante interações faz com que apenas os nós fiquem visíveis. Isso otimiza o processo de renderização enquanto o usuário interage com a GRN.
- **Utilização de cache de visualização para tornar a renderização durante a interação menos custosa:** utilizar esse recurso faz com que sejam geradas imagens do grafo, que simplificam o processo de renderização durante o movimento.

6 ANÁLISE DOS RESULTADOS

Para que o pesquisador fosse capaz de executar as ações planejadas na jornada do usuário, foram desenvolvidas algumas telas para separar as etapas do processo de construção e visualização de GRNs dentro do FunRegulation. A seguir, elas serão apresentadas e seus recursos explicados em detalhes.

A tela que inicia o processo de construção da GRN pode ser acessada através do menu *Build GRN*, localizado no topo da página. Ao abri-la, é disponibilizada uma caixa de seleção contendo todos os fungos disponíveis para construção da GRN, trazidos através da requisição apresentada no Capítulo 5.3.1, conforme visto na Figura 23. Após seleção do fungo desejado, são exibidas na tela as informações a seu respeito: ordem, gênero, espécie e cepa. Na Figura 24 é possível ver a apresentação das informações do fungo *Madurella mycetomatis*. Também é possível realizar a configuração da análise que será realizada. Duas opções estão disponíveis, Análise de Ortólogos utilizando a ferramenta *Protein Ortho* (44) e identificação dos Sítios de Ligação dos Fatores de Transcrição com a ferramenta *Regulatory Sequence Analysis Tools* (RSAT) (45), utilizada para identificar onde o TF se liga a região promotora do TG para ativar ou reprimir a expressão do gene. A primeira opção é obrigatória para todos os fungos, enquanto que o pesquisador pode escolher se irá utilizar ou não a segunda para a construção da GRN, caso o fungo possua matrizes de peso disponíveis. A Figura 25 apresenta as configurações da análise. Após realizar a escolha, clicando no botão *Build* será solicitada a construção da GRN.

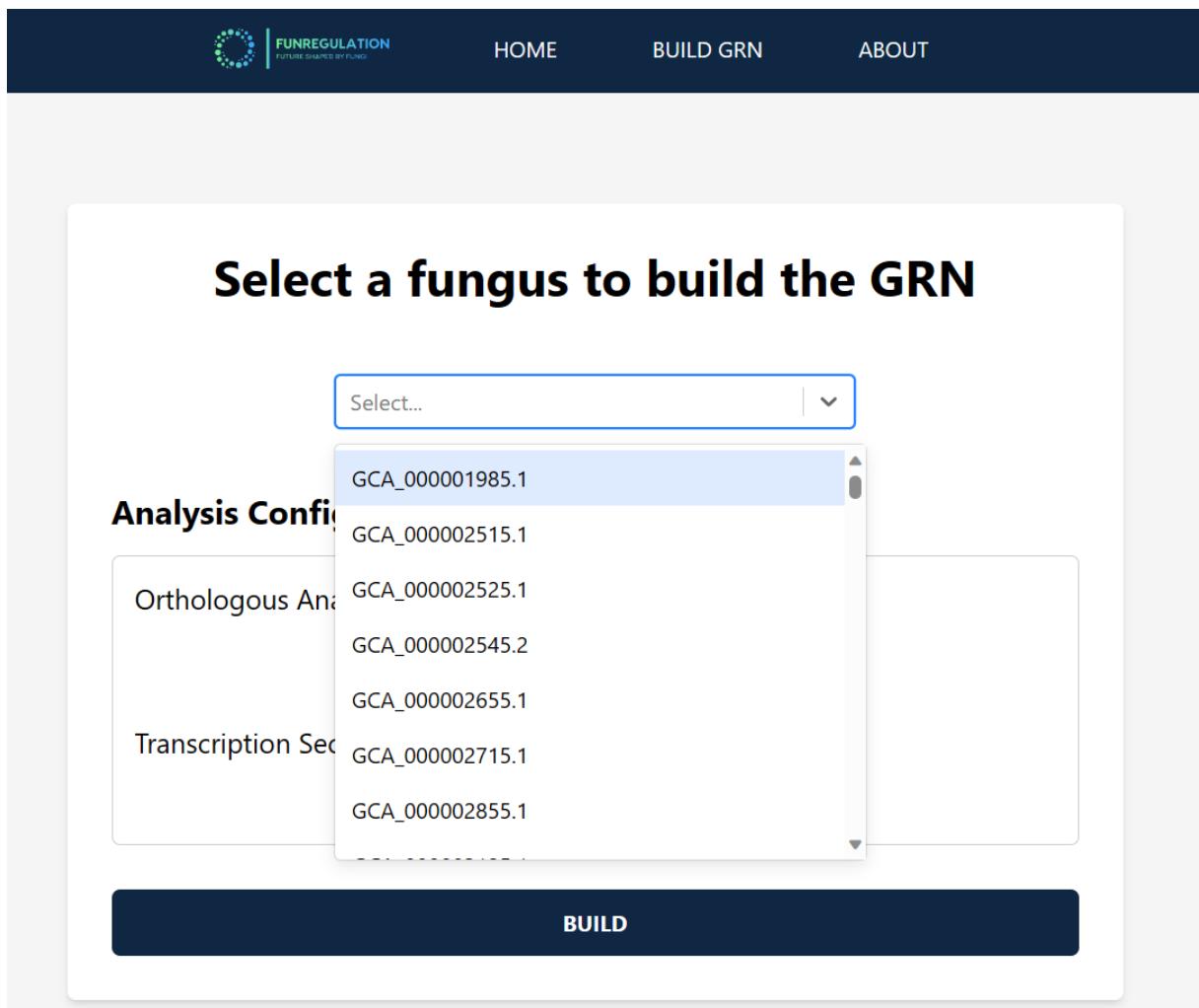


Figura 23 – Caixa de seleção do fungo para construção da GRN. Fonte: o autor.

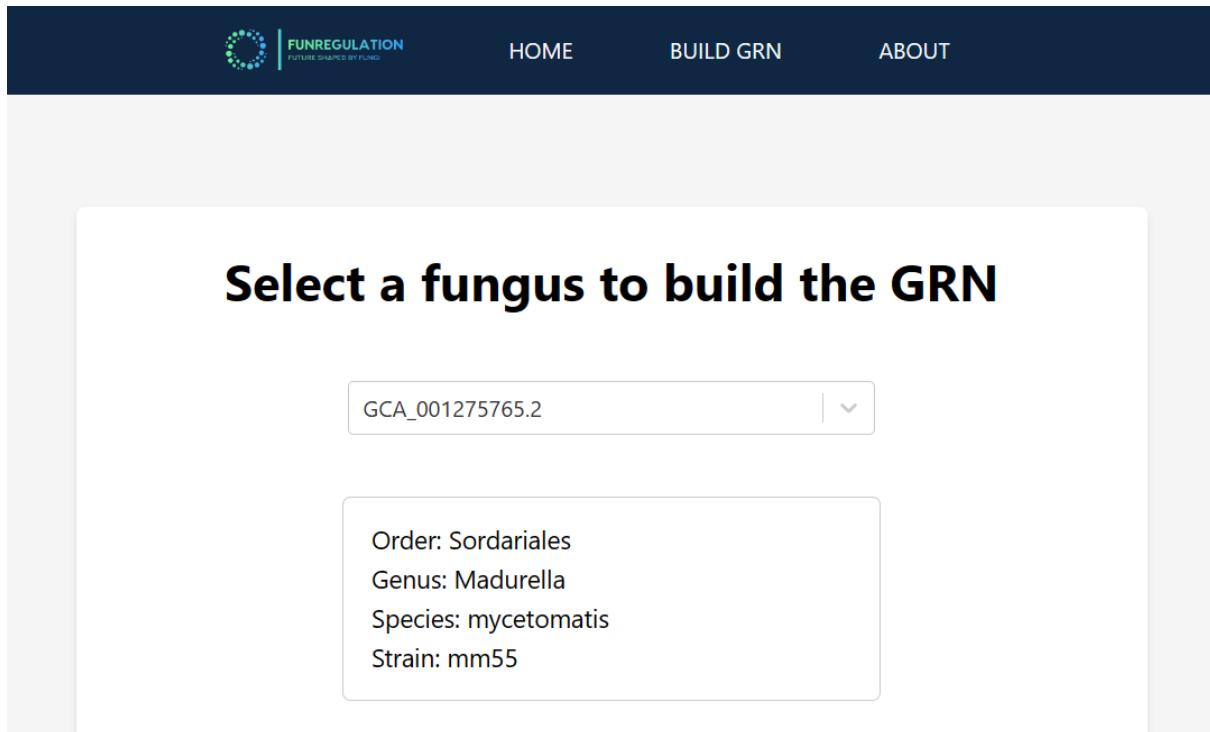


Figura 24 – Visualização de detalhes do fungo selecionado para construção da GRN. Fonte: o autor.

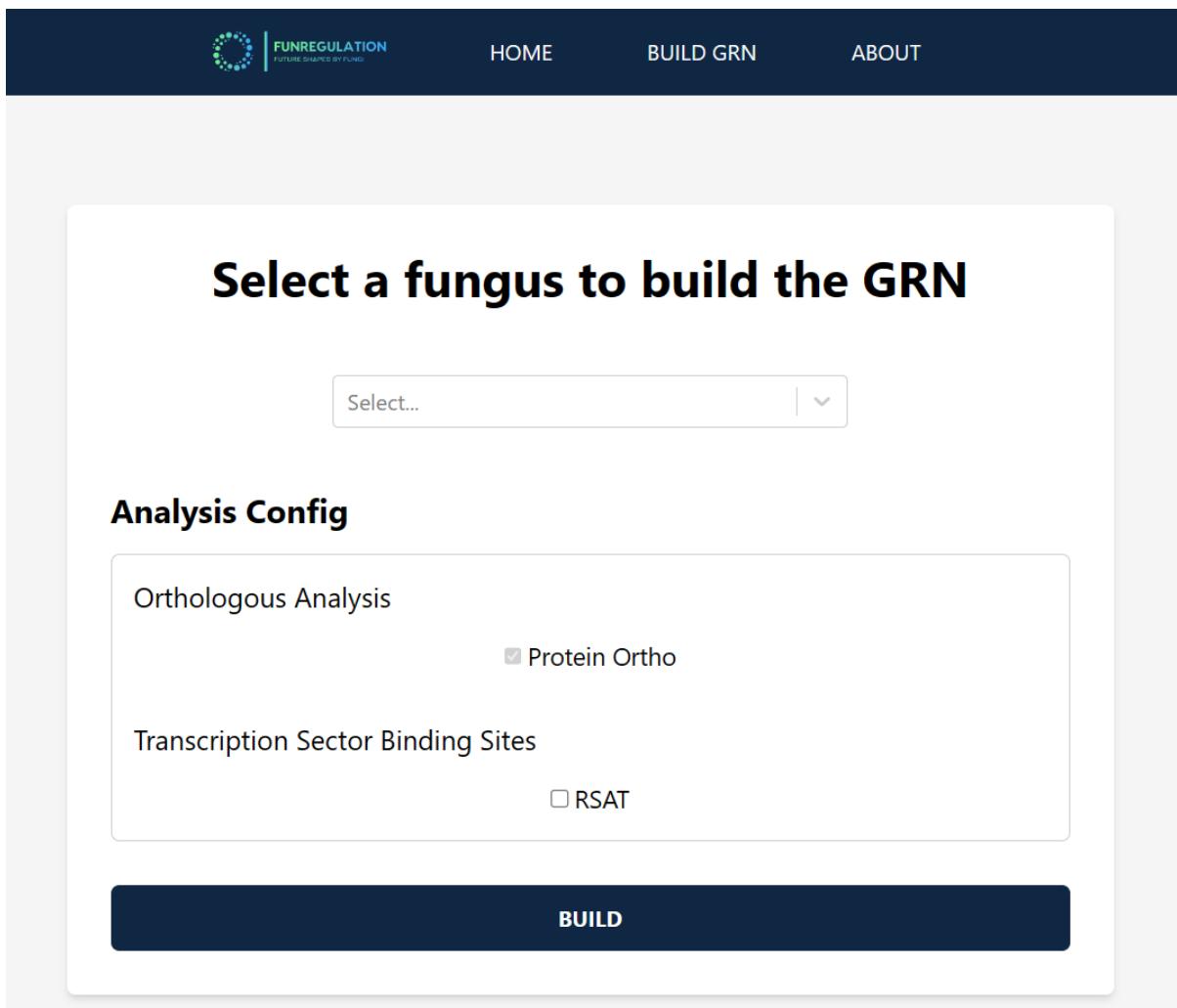


Figura 25 – Visão geral da tela de seleção do fungo para construção da GRN. Fonte: o autor.

Ao solicitar a construção da GRN, o pesquisador é redirecionado a uma página intermediária para acompanhar o status do processo. Todos eles estão inicialmente em vermelho, e após sua conclusão mudam para a cor verde. Quando todas as etapas do processo são finalizadas o botão *View GRN* é habilitado para que o pesquisador avance para a tela de visualização e manipulação da GRN. A Figura 26 apresenta a página de acompanhamento do status de construção da rede.

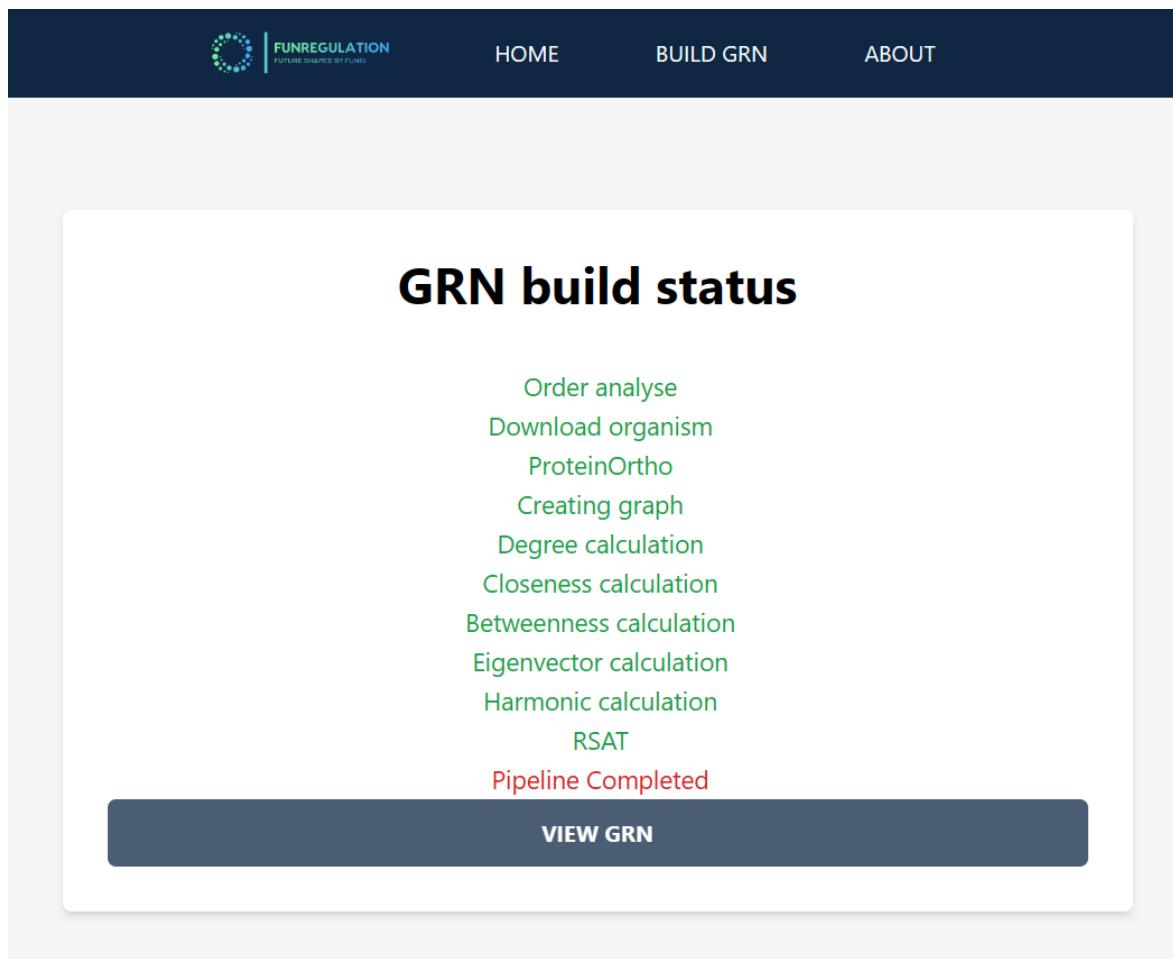


Figura 26 – Tela de acompanhamento do status de construção da GRN. Fonte: o autor.

Após clicar no botão *View GRN*, o pesquisador é direcionado para a tela de visualização da GRN construída. Aqui se encontra a maior parte das funcionalidades do sistema. Da esquerda para a direita, na parte de baixo da tela, a Figura 27 apresenta as funções disponíveis. Através dos botões *Zoom In* e *Zoom Out* é possível aumentar ou diminuir o zoom da visualização. Também é possível realizar essa ação com o botão de rolagem do mouse ou através do gesto de pinça com os dedos, caso o sistema esteja sendo utilizado em um *smartphone*. O botão *Circle* permite alterar o leiaute de visualização da GRN para o formato de círculo. Este formato pode ser visto na Figura 28. Como o grafo pode ser manipulado de diversas formas, através das funções disponíveis nos botões ou até mesmo selecionando e arrastando os nós individualmente, há disponível o botão *Restore*, que restaura a visualização da GRN para o estado inicial. Também é possível, para melhor visualização das interações de interesse do pesquisador, fazer o filtro dos TFs da GRN através da caixa de seleção ao lado direito do botão *Restore*. Ao selecionar os TFs de interesse e clicar no botão *Filter* ao lado direito, a visualização da GRN é alterada para

exibir apenas os nós que interagem com os TFs selecionados. A Figura 28 apresenta o filtro do TF MMYC01_200141, presente no fungo *Madurella mycetomatis* GCA_001275765.2. Ao lado do botão *Filter* há o botão *Export View*, que permite ao pesquisador exportar a visualização atual no formato de imagem. Quando o usuário clica no botão, o *download* da imagem é iniciado. A Figura 29 é uma imagem exportada a partir dessa funcionalidade. Por fim, há o botão *Hide Sidebar*, que ao ser selecionado esconde ou exibe a barra lateral que exibe informações sobre a GRN e o nó selecionado. Na barra lateral, são exibidas as estatísticas gerais da rede, que contém o número de interações, nós, autorregulações, interações positivas, interações negativas e interações desconhecidas, e as informações referentes ao nó selecionado pelo usuário, que contém *links* externos para visualização de dados do Interpro, EC, Cazy, PFAN, Panther, Genome3d, Uniprot, GO e Reactome. Na Figura 30 é possível ver as inforamções do nó CCM_00007, presente no fungo *Cordyceps militaris*.

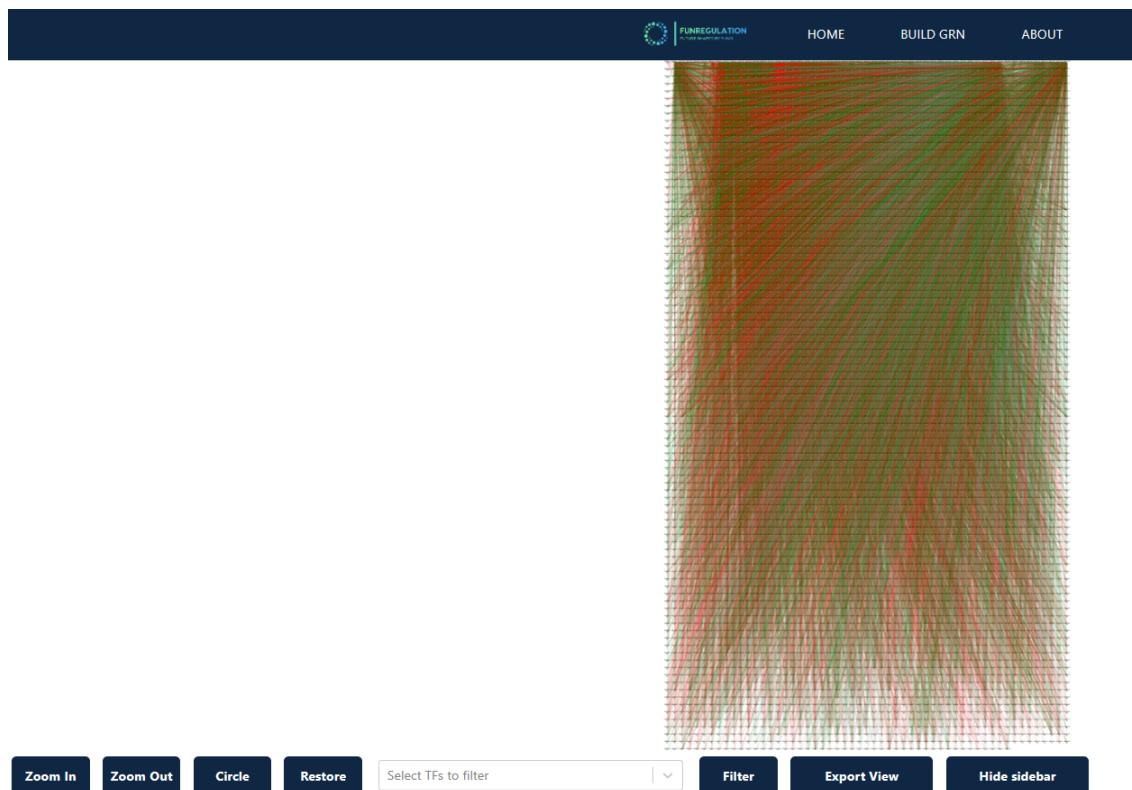


Figura 27 – Visualização da GRN construída para o *Madurella mycetomatis* GCA_001275765.2. Fonte: o autor.

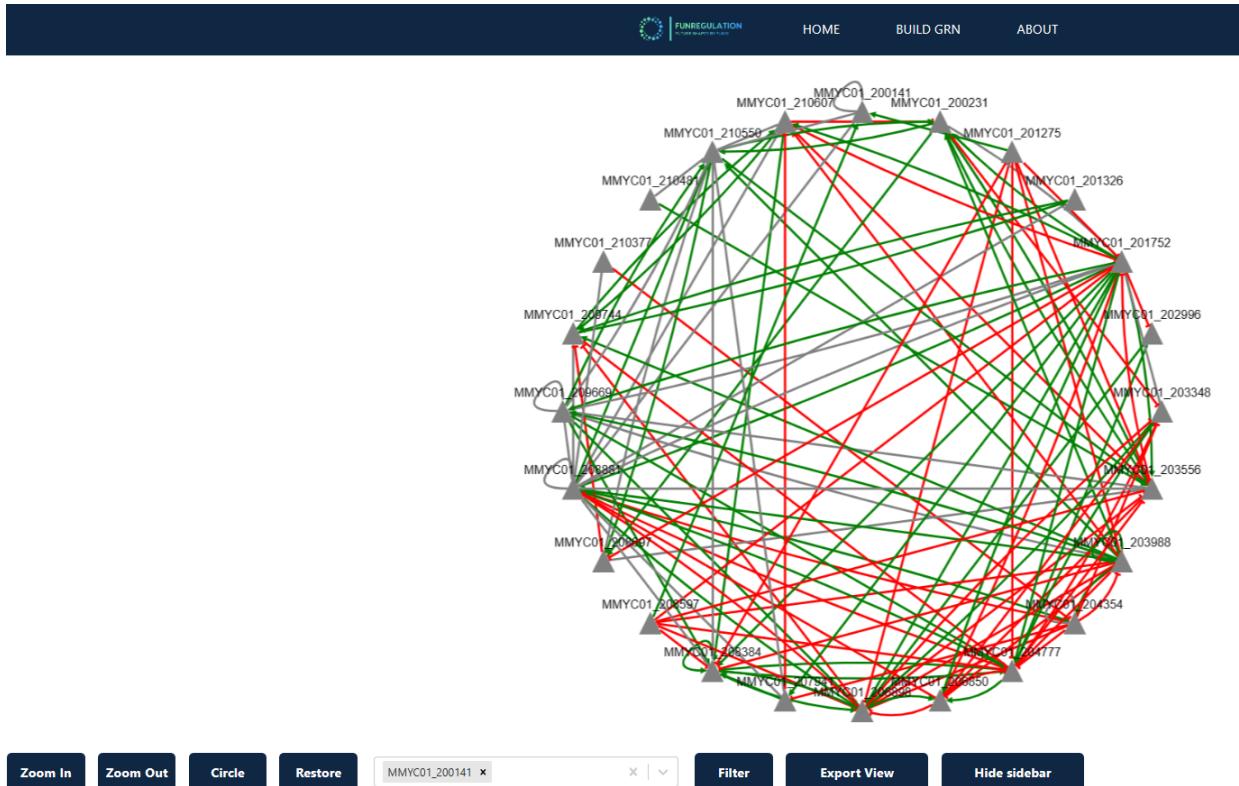


Figura 28 – Filtro do TF MMYC01_200141, presente no fungo *Madurella mycetomatis* GCA_001275765.2, e exibição de suas interações. Em verde as interações regulatórias positivas, em vermelho as negativas e em cinza as não identificadas. Fonte: o autor.

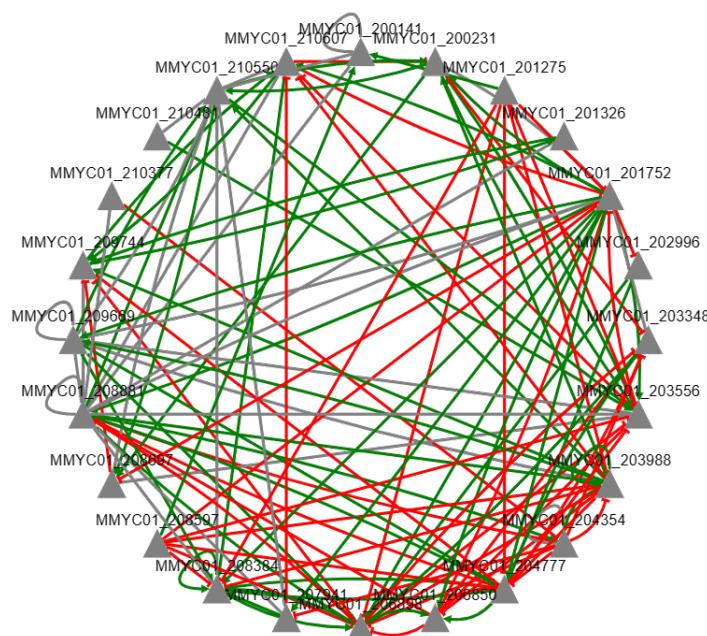


Figura 29 – Exemplo da imagem obtida através do botão *Export View*, exibindo o filtro do TF MMYC01_200141, presente no fungo *Madurella mycetomatis* GCA_001275765.2. Fonte: o autor.



Figura 30 – Visualização das estatísticas gerais e das informações do nó CCM_00007, presente no fungo *Cordyceps militaris* GCA_000225605-1. Fonte: o autor.

Para validar o funcionamento do *framework* Cytoscape na visualização e manipulação das GRNs, foram construídas duas GRNs. A primeira rede foi construída a partir do fungo *Madurella mycetomatis*, pode ser vista por completo na Figura 27 e ampliada na Figura 31, enquanto que a segunda foi construída a partir do fungo *Cordyceps militaris* e pode ser vista na Figura 32. A Tabela 1 apresenta as estatísticas da GRN para o primeiro fungo, enquanto que a Tabela 2 apresenta as estatísticas da GRN para o segundo.

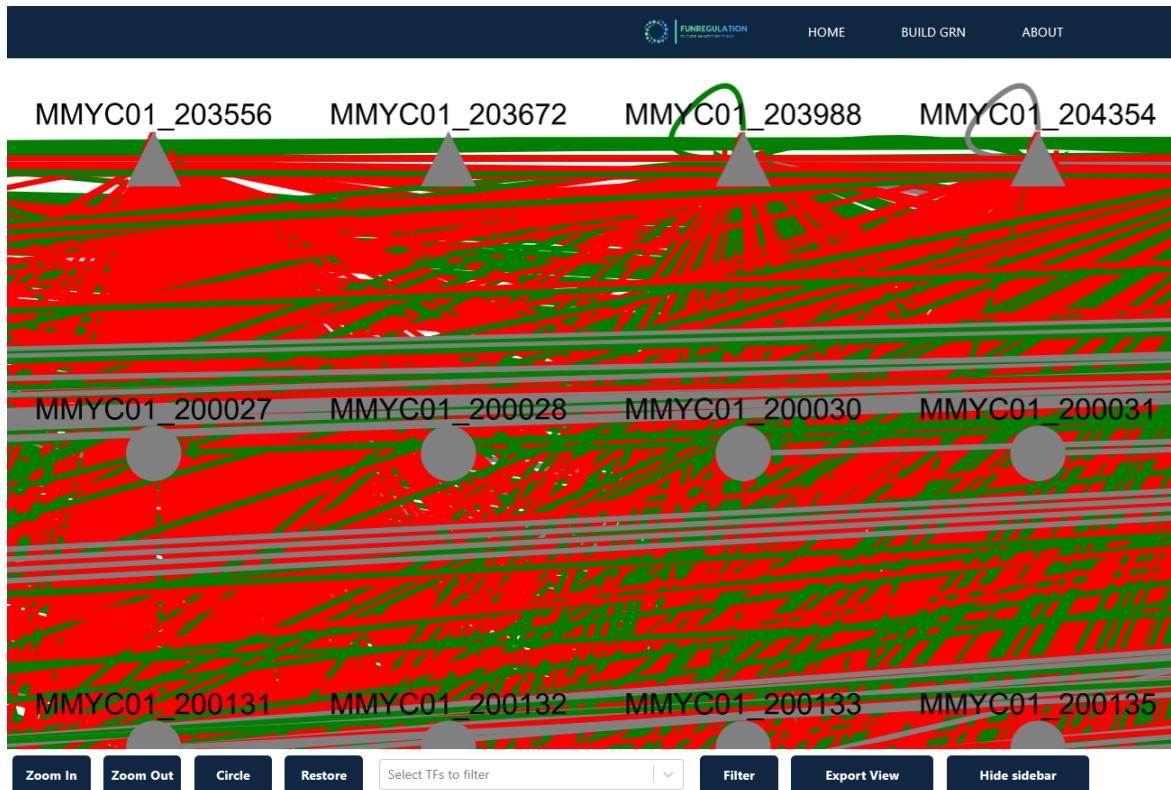


Figura 31 – Visualização da GRN construída para o *Madurella mycetomatis* GCA_001275765.2 ampliada. Fonte: o autor.

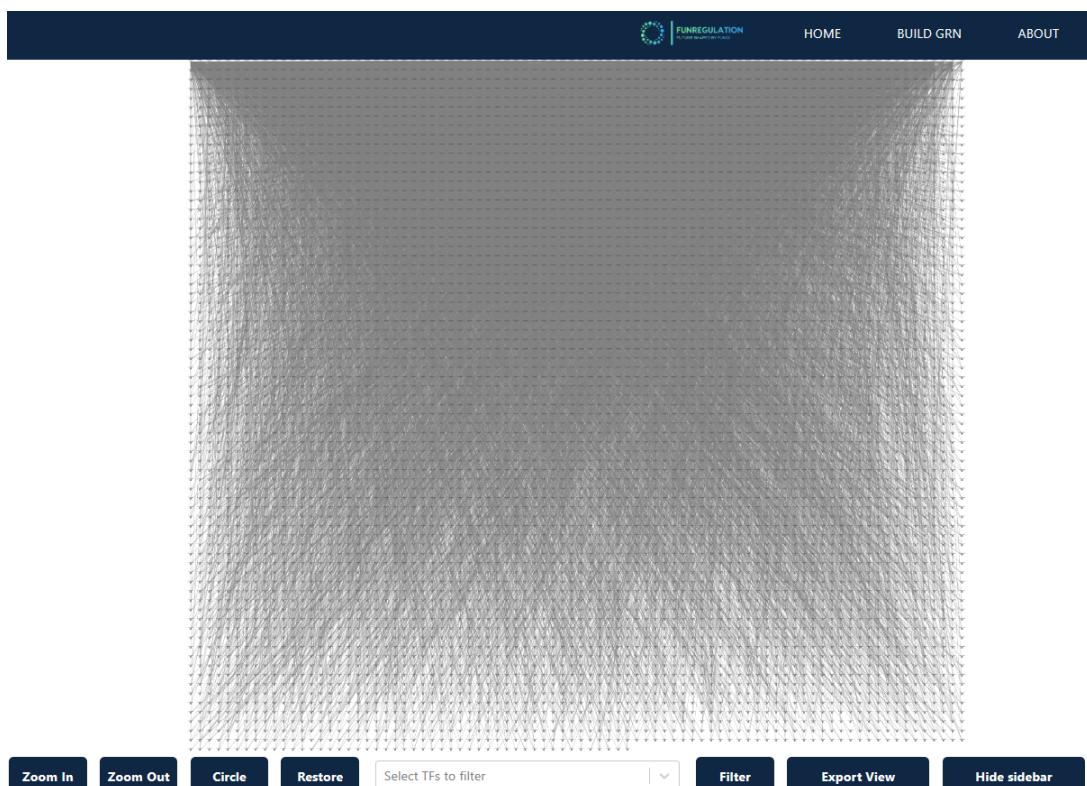


Figura 32 – Visualização da GRN construída para o *Cordyceps militaris* (GCA_000225605.1). Fonte: o autor.

Tabela 1 – Estatísticas da GRN construída para o fungo *Madurella mycetomatis*, da ordem Sordariales, cepa mm55 e de número de acesso GCA_001275765.2. Fonte: Adaptado de Gabriel Antonio Alves de Amorim.

Descrição	Valor
Interações	21.402
Nós	5.212
Autorregulações	28
Regulações positivas	9.181
Regulações negativas	8.373
Regulações desconhecidas	3.848

Tabela 2 – Estatísticas da GRN construída para o fungo *Cordyceps militaris*, da ordem Hypocreales, cepa CM01 e com o número de acesso GCA_000225605.1. Fonte: Adaptado de Gabriel Antonio Alves de Amorim.

Descrição	Valor
Interações	14.847
Nós	6.345
Autorregulações	0
Regulações positivas	0
Regulações negativas	0
Regulações desconhecidas	14.847

Para apurar se o carregamento da GRN é eficiente em termos de tempo e memória, foram realizados testes de carregamento das duas GRNs construídas pelo sistema. Para realização dos testes, foi utilizado um computador com as seguintes especificações: processador Intel Core i5-1235U, memória RAM de 16GB DDR4 3200Mhz e sistema operacional Windows 11 Home 64 bits. O ambiente de testes foi preparado de maneira que apenas o editor de código Visual Studio Code executando o servidor da aplicação, o programa Bloco de Notas para coleta dos dados e o navegador Edge estivessem abertos, para evitar qualquer interferência externa nos resultados. Os valores coletados foram o *timestamp* no início do carregamento da GRN, o *timestamp* no fim do carregamento da GRN, ambos coletados através de variáveis que armazenaram os valores obtidos através do método *getTime* (46), presente no objeto *Date* (47) do JavaScript, e o tamanho do *heap* JavaScript, presente na aba Monitor de Desempenho do navegador Edge, que apresenta a quantidade de memória utilizada pelo programa JavaScript na página (48). Para evitar que o navegador utilizasse dados armazenados no cache da página, o FunRegulation foi aberto em uma guia anônima do navegador Edge e ao final de cada teste, o navegador era fechado e reaberto para que o novo teste fosse executado em um ambiente completamente limpo. O teste foi executado

três vezes para cada uma das GRNs e consistia nos seguintes passos:

- Abertura da página de seleção de fungo.
- Escolha do fungo desejado.
- Clique no botão *Build*.
- Avanço para a página de acompanhamento do status de construção da GRN.
- Clique no botão *View GRN*.
- Avanço para a página de visualização e espera do carregamento da rede.
- Abertura do console do navegador e coleta dos timestamps do início e fim do carregamento da GRN.
- Abertura da aba Monitor de Desempenho e coleta do tamanho do *heap* de memória informado.

As tabelas 3 e 4 apresentam os resultados dos testes realizados para o fungo *Madurella mycetomatis* e *Cordyceps militaris*, respectivamente. Para o cálculo do tempo do carregamento das GRNs em segundos, foi realizada a seguinte operação: `timestamp_fim - timestamp_inicio`.

Tabela 3 – Resultado dos testes de carregamento realizados para a GRN do fungo *Madurella mycetomatis*, da ordem Sordariales, cepa mm55 e de número de acesso GCA_001275765.2. Fonte: o autor.

Nº teste	<code>timestamp_inicio</code>	<code>timestamp_fim</code>	Tempo em segundos	Tam. heap de memória
1	1701390255531	1701390259843	4,312s	168MB
2	1701390370275	1701390374148	3,873s	166MB
3	1701390426229	1701390430733	4,504s	165MB

Tabela 4 – Resultado dos testes de carregamento realizados para a GRN do fungo *Cordyceps militaris*, cepa CM01 e com o número de acesso GCA_000225605.1. Fonte: o autor.

Nº teste	<code>timestamp_inicio</code>	<code>timestamp_fim</code>	Tempo em segundos	Tam. heap de memória
1	1701390487448	1701390489420	1,972s	168MB
2	1701390535801	1701390537770	1,969s	164MB
3	1701390587749	1701390589712	1,963s	166MB

A média de tempo para carregamento da GRN do *Madurella mycetomatis* foi de 4.230 segundos enquanto que para o *Cordyceps militaris* 1.968 segundos. A diferença de tempo pode estar relacionada com o fato de que no segundo fungo há menos interações regulatórias, conforme pode ser visto na Tabela 2, o que torna a renderização do grafo menos custosa para o

computador. Também há o fato de que o tipo das interações regulatórias não foram identificados, o que também pode contribuir para o carregamento mais rápido, já que há menos variações no momento da renderização do grafo. Tais afirmações são ditas com base na análise dos dados coletados e carecem de testes para confirmação, que podem ser realizados em um trabalho futuro. A média de utilização de memória durante o carregamento da rede para o primeiro fungo foi de 166,33MB, enquanto que para o segundo foi de 166MB. Neste ponto, não houve uma diferença significativa entre as duas GRNs.

É importante lembrar que o teste foi realizado em um ambiente local, na máquina descrita. As requisições para a API foram simuladas com dados reais coletados dos *endpoints* especificados no Capítulo 5.3 em um momento anterior e não realizadas diretamente ao *back-end* do *FunRegulation* no instante do teste. Por conta disso, o tempo de resposta da requisição obtido, que influencia no tempo total de carregamento da GRN, será menor do que se o teste fosse realizado com o sistema hospedado em um servidor na Internet.

7 CONSIDERAÇÕES FINAIS

A construção da aplicação Web *front-end FunRegulation*, para visualização e manipulação de GRNs de fungos contribui para que seja possível entender a relação entre genes nesses seres vivos. O sistema remove a necessidade de realizar configurações complexas do ambiente para construção e visualização de GRNs, ao permitir que o pesquisador realize essa tarefa através de um navegador Web. Junto a isso, também é eliminada a necessidade de haver grande poder de processamento e memória no computador do usuário, já que o processo de construção da GRN é de responsabilidade do *back-end* do *FunRegulation*, restando apenas a tarefa de renderização do grafo para o computador no momento em que ocorre a visualização e manipulação da rede, também facilitada tendo em vista que a solução seguiu as boas práticas recomendadas pelas ferramentas utilizadas no desenvolvimento deste trabalho, descritas no Capítulo 2.2.

O projeto da jornada do usuário, um dos objetivos específicos deste trabalho e descrita no Capítulo 5.1, contribuiu para a decisão da disposição das funcionalidades construídas para o sistema. A realização dos testes de carregamento das duas GRNs geradas, descritos no Capítulo 6, demonstrou a funcionalidade da arquitetura para visualização de redes construída com o Cytoscape.js e descrita no Capítulo 5.2, e também mostrou a eficiência de carregamento das GRNs em termos de tempo e memória. Desta forma, considera-se que todos os objetivos específicos propostos para a elaboração deste trabalho foram alcançados.

Entre os trabalhos futuros identificados durante a execução deste projeto estão:

- Implementação dos casos de uso restantes que se referem ao *front-end* do *FunRegulation*.
- Finalização da integração entre *front-end*. *back-end* da aplicação, já que atualmente os dados solicitados através dos *endpoints* estão no formato fornecido pelo *back-end*, porém não são provenientes dele em tempo real.
- Disponibilização da ferramenta na Web para a comunidade científica.
- Avaliação da contribuição com a comunidade através de entrevistas ou questionários, conforme estipulado na avaliação 2 do artefato, descrito no capítulo 3.3.
- Identificação e correção de possíveis *bugs* encontrados após publicação da ferramenta.

REFERÊNCIAS

- 1 LENZ, A. R. **O Projeto Genoma de *Penicillium echinulatum* 2HH e S1M29: A Genômica Viabilizando a Descoberta de Conhecimento.** 2020.
- 2 JENSEN, R. A. Orthologs and paralogs - we need to get it right. **Genome Biology**, v. 2, n. 8, p. INTERACTIONS1002, 2001. ISSN 1474-760X.
- 3 ZENG, Y.; GONG, M.; LIN, M.; GAO, D.; ZHANG, Y. A Review About Transcription Factor Binding Sites Prediction Based on Deep Learning. **IEEE Access**, v. 8, p. 219256–219274, 2020. ISSN 2169-3536. Conference Name: IEEE Access. Disponível em: <<https://ieeexplore.ieee.org/document/9284502>>.
- 4 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>>.
- 5 COLA.JS/Cytoscape.JS. Disponível em: <<https://js.cytoscape.org/demos/colajs-graph/>>.
- 6 PIMENTEL, M.; FILIPPO, D.; SANTOS, T. M. Design science research: pesquisa científica atrelada ao design de artefatos. **RE@D – Revista de Educação a Distância e eLearning**, p. 37–61, abr. 2020. ISSN 2182-4967. Accepted: 2020-06-16T11:16:11Z Publisher: Universidade Aberta. Laboratório de Educação a Distância e Elearning (LE@D). Disponível em: <<https://repositorioaberto.uab.pt/handle/10400.2/9787>>.
- 7 WHAT is Scrum? | Scrum.org. Disponível em: <<https://www.scrum.org/resources/what-scrum-module>>.
- 8 YADAV, D. Relevance of Bioinformatics in the Era of Omics Driven Research. **Journal of Next Generation Sequencing & Applications**, v. 02, n. 01, 2015. ISSN 24699853. Disponível em: <<https://www.omicsonline.org/open-access/relevance-of-bioinformatics-in-the-era-of-omics-driven-research-jngsa-1000e102.php?aid=56383>>.

- 9 NOTARI, D. L.; ALBA, G. D.; SILVA, S. d. A. e. **Bioinformática: Contexto Computacional e Aplicações**. 1. ed. [s.n.], 2020. ISBN 978-65-5807-001-6. Disponível em: <<https://www.ucs.br/educs/livro/bioinformatica-contexto-computacional-e-aplicacoes/>>.
- 10 MOMPTEL, P. Badia-i; WESSELS, L.; MÜLLER-DOTT, S.; TRIMBOUR, R.; FLORES, R. O. R.; ARGELAGUET, R.; SAEZ-RODRIGUEZ, J. Gene regulatory network inference in the era of single-cell multi-omics. **Nature Reviews Genetics**, v. 24, n. 11, p. 739–754, nov. 2023. ISSN 1471-0064. Number: 11 Publisher: Nature Publishing Group. Disponível em: <<https://www.nature.com/articles/s41576-023-00618-5>>.
- 11 TRANSCRIPTION. Disponível em: <<https://www.genome.gov/genetics-glossary/Transcription>>.
- 12 GENE Regulation. Disponível em: <<https://www.genome.gov/genetics-glossary/Gene-Regulation>>.
- 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 H, X.; H, Y.; K, T.; Q, S.; C, W.; YY, L.; YX, L. 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**, v. 7 Suppl 2, n. Suppl 2, 2013. ISSN 1752-0509. Disponível em: <<https://pubmed.ncbi.nlm.nih.gov/24565134/>>.
- 15 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**, v. 9, p. e51254, jan. 2020. ISSN 2050-084X. Publisher: eLife Sciences Publications, Ltd. Disponível em: <<https://doi.org/10.7554/eLife.51254>>.
- 16 GABALDÓN, T.; KOONIN, E. V. Functional and evolutionary implications of gene orthology. **Nature Reviews Genetics**, v. 14, n. 5, p. 360–366, maio 2013. ISSN 1471-0064. Number: 5 Publisher: Nature Publishing Group. Disponível em: <<https://www.nature.com/articles/nrg3456>>.
- 17 TAN, K.; SHLOMI, T.; FEIZI, H.; IDEKER, T.; SHARAN, R. Transcriptional regulation of protein complexes within and across species. **Proceedings of the National Academy of Sciences**, v. 104, n. 4, p. 1283–1288, jan. 2007. ISSN 0027-8424, 1091-6490. Disponível em: <<https://pnas.org/doi/full/10.1073/pnas.0606914104>>.
- 18 ZOGOPOULOS, V. L.; SPAHO, K.; NTOKA, C.; LAPPAS, G. A.; KYRANIS, I.; BAGOS, P. G.; SPANDIDOS, D. A.; MICHALOPOULOS, I. TFBSPred: A functional transcription factor binding site prediction webtool for humans and mice. **International Journal of Epigenetics**, v. 1, n. 4, p. 1–11, nov. 2021. ISSN 2752-5406. Publisher: Spandidos Publications. Disponível em: <<https://www.spandidos-publications.com/10.3892/ije.2021.9>>.
- 19 DOCS | Next.js. Disponível em: <<https://nextjs.org/docs>>.
- 20 API Reference: create-next-app | Next.js. Disponível em: <<https://nextjs.org/docs/app/api-reference/create-next-app>>.

- 21 WALTON, P. **Métricas da Web | Articles**. Disponível em: <<https://web.dev/articles/vitals?hl=pt-br>>.
- 22 NODE.JS. Disponível em: <<https://nodejs.org/en/>>.
- 23 GETTING Started: Project Structure | Next.js. Disponível em: <<https://nextjs.org/docs/getting-started/project-structure>>.
- 24 ABOUT Cytoscape.js. Disponível em: <<https://js.cytoscape.org/#introduction/about>>.
- 25 CYTOSCAPE.JS Architecture & API. Disponível em: <<https://js.cytoscape.org/#notation/architecture-and-api>>.
- 26 ECMA-404. Disponível em: <<https://ecma-international.org/publications-and-standards/standards/ecma-404/>>.
- 27 FRANZ, M.; LOPES, C. T.; HUCK, G.; DONG, Y.; SUMER, O.; BADER, G. D. Cytoscape.js: a graph theory library for visualisation and analysis. **Bioinformatics**, v. 32, n. 2, p. 309–311, jan. 2016. ISSN 1367-4811, 1367-4803. Disponível em: <<https://academic.oup.com/bioinformatics/article/32/2/309/1744007>>.
- 28 CSS | MDN. 2022. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/CSS>>.
- 29 UTILITY-First Fundamentals - Tailwind CSS. Disponível em: <<https://tailwindcss.com/docs/utility-first>>.
- 30 A comprehensive guide to design systems | Inside Design Blog. Disponível em: <<https://www.invisionapp.com/inside-design/guide-to-design-systems/>>.
- 31 INSTALLATION - Tailwind CSS. Disponível em: <<https://tailwindcss.com/docs/installation>>.
- 32 LENZ, A. R.; GALÁN-VÁSQUEZ, E.; BALBINOT, E.; ABREU, F. P. de; OLIVEIRA, N. Souza de; ROSA, L. O. da; SILVA, S. de A. 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, p. 588263, 2020. ISSN 1664-302X.
- 33 DÍAZ-MONTAÑA, J. J.; DÍAZ-DÍAZ, N.; BARRANCO, C. D.; PONZONI, I. Development and use of a Cytoscape app for GRNCOP2. **Computer Methods and Programs in Biomedicine**, v. 177, p. 211–218, ago. 2019. ISSN 1872-7565.
- 34 WEINANDT, N.; JACKSON, L. M.; GNIMPIEBA, E. Z.; LUSHBOUGH, C. 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**. New York, NY, USA: Association for Computing Machinery, 2014. (BCB '14), p. 797–804. ISBN 978-1-4503-2894-4. Disponível em: <<http://doi.org/10.1145/2649387.2660851>>.
- 35 GUAN, D.; SHAO, J.; ZHAO, Z.; WANG, P.; QIN, J.; DENG, Y.; BOHELER, K. R.; WANG, J.; YAN, B. PTHGRN: unraveling post-translational hierarchical gene regulatory networks using PPI, ChIP-seq and gene expression data. **Nucleic Acids Research**, v. 42, n. Web Server issue, p. W130–136, jul. 2014. ISSN 1362-4962.

- 36 GIL, A. C. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2007.
- 37 HOME | Scrum Guides. Disponível em: <<https://scrumguides.org/>>.
- 38 KANBAN and Scrum - Making the Most of Both. Disponível em: <<https://www.infoq.com/minibooks/kanban-scrum-minibook/>>.
- 39 AHMAD, M. O.; MARKKULA, J.; OIVO, M. Kanban in software development: A systematic literature review. In: **2013 39th Euromicro Conference on Software Engineering and Advanced Applications**. [s.n.], 2013. p. 9–16. ISSN: 2376-9505. Disponível em: <<https://ieeexplore.ieee.org/abstract/document/6619482>>.
- 40 MANAGE Your Team's Projects From Anywhere | Trello. Disponível em: <<https://trello.com/home>>.
- 41 COSENTINO, V.; IZQUIERDO, J. L. C.; CABOT, J. A Systematic Mapping Study of Software Development With GitHub. **IEEE Access**, v. 5, p. 7173–7192, 2017. ISSN 2169-3536. Conference Name: IEEE Access. Disponível em: <<https://ieeexplore.ieee.org/abstract/document/7887704>>.
- 42 LENZ, A.; VIEIRA, S. **G2BC/funregulation-frontend**. Disponível em: <<https://github.com/G2BC/funregulation-frontend>>.
- 43 Códigos de status de respostas HTTP - HTTP | MDN. 2023. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTTP>Status>>.
- 44 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, abr. 2011. ISSN 1471-2105. Disponível em: <<https://doi.org/10.1186/1471-2105-12-124>>.
- 45 SANTANA-GARCIA, W.; CASTRO-MONDRAGON, J. A.; PADILLA-GÁLVEZ, M.; NGUYEN, N. T. T.; ELIZONDO-SALAS, A.; KSOURI, N.; GERBES, F.; THIEFFRY, D.; VINCENS, P.; CONTRERAS-MOREIRA, B.; HELDEN, J. van; THOMAS-CHOLLIER, M.; MEDINA-RIVERA, A. RSAT 2022: regulatory sequence analysis tools. **Nucleic Acids Research**, v. 50, n. W1, p. W670–W676, jul. 2022. ISSN 0305-1048. Disponível em: <<https://doi.org/10.1093/nar/gkac312>>.
- 46 DATE.PROTOTYPE.getTime() - JavaScript | MDN. 2023. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date/getTime>.
- 47 DATE - JavaScript | MDN. 2023. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date>.
- 48 MSEdgeTEAM. **Medir o desempenho do runtime de uma página usando a ferramenta Monitor de Desempenho - Microsoft Edge Development**. 2023. Disponível em: <<https://learn.microsoft.com/pt-br/microsoft-edge/devtools-guide-chromium/performance-monitor/performance-monitor-tool>>.

APÊNDICES

APÊNDICE A – Documento de Requisitos

	<p>FunRegulation Construção e Manipulação de Redes Regulatórias de Genes de Fungos</p>	 <p>UNEBA <small>UNIVERSIDADE DO ESTADO DA BAHIA</small></p>
<p>Documento de Definição de Requisitos</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

Salvador
2021

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

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

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

	de funções para cada gene de um genoma.
Symbol gene	Abreviação do nome de um determinado gene.
Locus tag	Identificadores aplicados sistematicamente a cada gene em um projeto de sequenciamento.
TF	Fator de Transcrição (Transcription Factor) - proteínas que ajudam a regular a transcrição de TGs através da conexão ao DNA próximo ao TG.
TG	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		
Descrição:	O sistema deve permitir o cadastro de usuários a partir dos seguintes campos: <ul style="list-style-type: none">● Nome completo;● E-mail institucional;● Senha;● Nome da instituição;● País;● Se for um usuário brasileiro, selecionar o estado.	
Origem:	Alexandre Lenz	
Modificado por:	Leandro Lopes	
Versão: 2.0	Status: Aprovado	Prioridade: Média
RNs	RN02; RN03;	

RF02 - Confirmar cadastro	
Descrição:	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.

Origem:	Alexandre Lenz	
Modificado por:	Leandro Lopes	
Versão: 2.0	Status: Aprovado	Prioridade: Média

RF03 – Efetuar Log in		
Descrição:	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.	
Origem:	Alexandre Lenz	
Modificado por:	Leandro Lopes	
Versão: 2.0	Status: Aprovado	Prioridade: Média

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

RF05 - Recuperar senha

Descrição:	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.	
Origem:	Alexandre Lenz	
Modificado por:	Leandro Lopes	
Versão: 2.0	Status: Aprovado	Prioridade: Média
RNs	RN01	

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

	<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>	
Origem:	Alexandre Lenz	
Modificado por:	Leandro Lopes	
Versão: 2.0	Status: Aprovado	Prioridade: Alta

RF07 – Carregar informações genômicas de Ensembl Fungi		
Descrição:	<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"> ● Genoma completo no formato fasta ● Proteínas no formato fasta ● Anotação genômica no formato gff3 ● Anotação funcional das proteínas contendo GO terms e/ou InterPro 	
Origem:	Alexandre Lenz	
Modificado por:	Leandro Lopes	
Versão: 2.0	Status: Aprovado	Prioridade: Média
RNs	RN01; RN04; RN05; RN08	

RF08 - Carregar informações genômicas de arquivos locais

Descrição:	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"> ● Genoma completo no formato fasta ● Proteínas no formato fasta ● Anotação genômica no formato gff3 ● Anotação funcional no formato tsv (deve conter InterPro e/ou GO terms) 	
Origem:	Alexandre Lenz	
Modificado por:	Leandro Lopes	
Versão: 2.0	Status: Aprovado	Prioridade: Alta
Rastreamento	RN01; RN05.	

RF09 – Exibir e Manipular GRN

Descrição:	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"> ● O usuário poderá selecionar um nó da GRN para ver sua anotação funcional, conforme RF10; ● O usuário poderá exibir estatísticas gerais da GRN, conforme RF11; 	
Origem:	Alexandre Lenz	
Modificado por:		
Versão: 1.0	Status: Aprovado	Prioridade: Alta
RNs	RN06; RN07;	

RF10 – Exibir anotação funcional de um nó

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

RF11 - Exibir estatísticas

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

RF12 – Configurar GRN

Descrição:	O sistema deve permitir que o usuário configure a GRN conforme suas expectativas, de acordo com a RN08: <ul style="list-style-type: none"> ● 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; ● O usuário poderá ampliar ou reduzir o tamanho da GRN, conforme RF13; ● O usuário poderá escolher o layout de exibição da GRN, conforme RF14; ● O usuário poderá filtrar quais interações regulatórias devem ser mostradas na GRN, conforme RF15;
-------------------	---

Origem:	Alexandre Lenz	
Modificado por:		
Versão: 1.0	Status: Aprovado	Prioridade: Baixa
RNs	RN08;	

RF13 - Permitir zoom		
Descrição:	<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"> ● Scroll do mouse; ● Pinça no touchpad; ● teclas de atalho + e -; ● botões +, - e fit-to-window. 	
Origem:	Alexandre Lenz	
Modificado por:	Jéssica Cardoso	
Versão: 2.0	Status: Aprovado	Prioridade: Baixa

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

--	--	--

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

RF16 – Manter GRNs e Views		
Descrição:	<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>	
Origem:	Alexandre Lenz	
Modificado por:	Jéssica Cardoso	
Versão: 2.0	Status: Aprovado	Prioridade: Alta

RF17 - Exportar GRN ou View

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

RF18 - Mostrar tutorial

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

RF19 - Mostrar tela inicial

Descrição:	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">● E-mail para problemas científicos;● E-mail para problemas técnicos;● Citação do artigo publicado.	
Origem:	Alexandre Lenz	
Modificado por:		
Versão: 1.0	Status: Aprovado	Prioridade: Baixa

4.2 Requisitos Não-Funcionais

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

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

Critérios de Aceitação:	Todos os organismos da plataforma Ensembl Fungi devem estar disponíveis no sistema, respeitando RN-4.
--------------------------------	---

RNF02 - Usuários Simultâneos		
Descrição:	O sistema deverá suportar processamento multiusuário, ou seja, suportar no máximo 10 usuários conectados e operando o sistema simultaneamente.	
Versão: 2.0	Status: Aprovado	Prioridade: Média
Origem:	Jéssica Cardoso	
Modificado por:	Jéssica Cardoso	
Tipo:	Performance	
Critérios de Aceitação:	Teste de Acesso: O sistema deve permitir 10 usuários simultâneos, sendo que acima dessa quantidade os usuários que tentarem efetuar o login serão alertados que devem esperar até que a carga do sistema diminua.	

RNF03 - Tempo de Resposta		
Descrição:	Cada consulta ou interação com o sistema não deve ultrapassar dez segundos.	
Versão: 2.0	Status: Aprovado	Prioridade: Média

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

RNF04 - Log auditável		
Descrição:	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.	
Versão: 2.0	Status: Aprovado	Prioridade: Média
Origem:	Alexandre Lenz	
Modificado por:	Alexandre Lenz	
Tipo:	Manutenibilidade	
Critérios de Aceitação:	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		
Descrição:	RFs 06 a 17 somente poderão ser executados por usuários logados no sistema.	
Versão: 1.0	Status: Aprovado	Prioridade: Média
Origem:	Alexandre Lenz e Kevin Lima	
Modificado por:		

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

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

RN04 - Importar organismos

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

RN05 - Upload de arquivos

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

RN06 - Identificar elementos da GRN

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

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

Modificado por:	Alexandre Lenz
------------------------	----------------

RN08 – Salvar configurações automaticamente		
Descrição:	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.	
Versão: 2.0	Status: Aprovado	Prioridade: Média
Origem:	Alexandre Lenz	
Modificado por:	Alexandre Lenz	

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

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 conflituante?
- Todos requisitos estão descritos de forma clara e objetiva?
- Os requisitos podem ser verificados?
- Os requisitos são rastreáveis?

APÊNDICE B – Documento de Casos de Uso

	<p style="text-align: center;">FunRegulation Construção e Manipulação de Redes Regulatórias de Genes de Fungos</p>	
<p>Documento de Análise de Requisitos</p>		

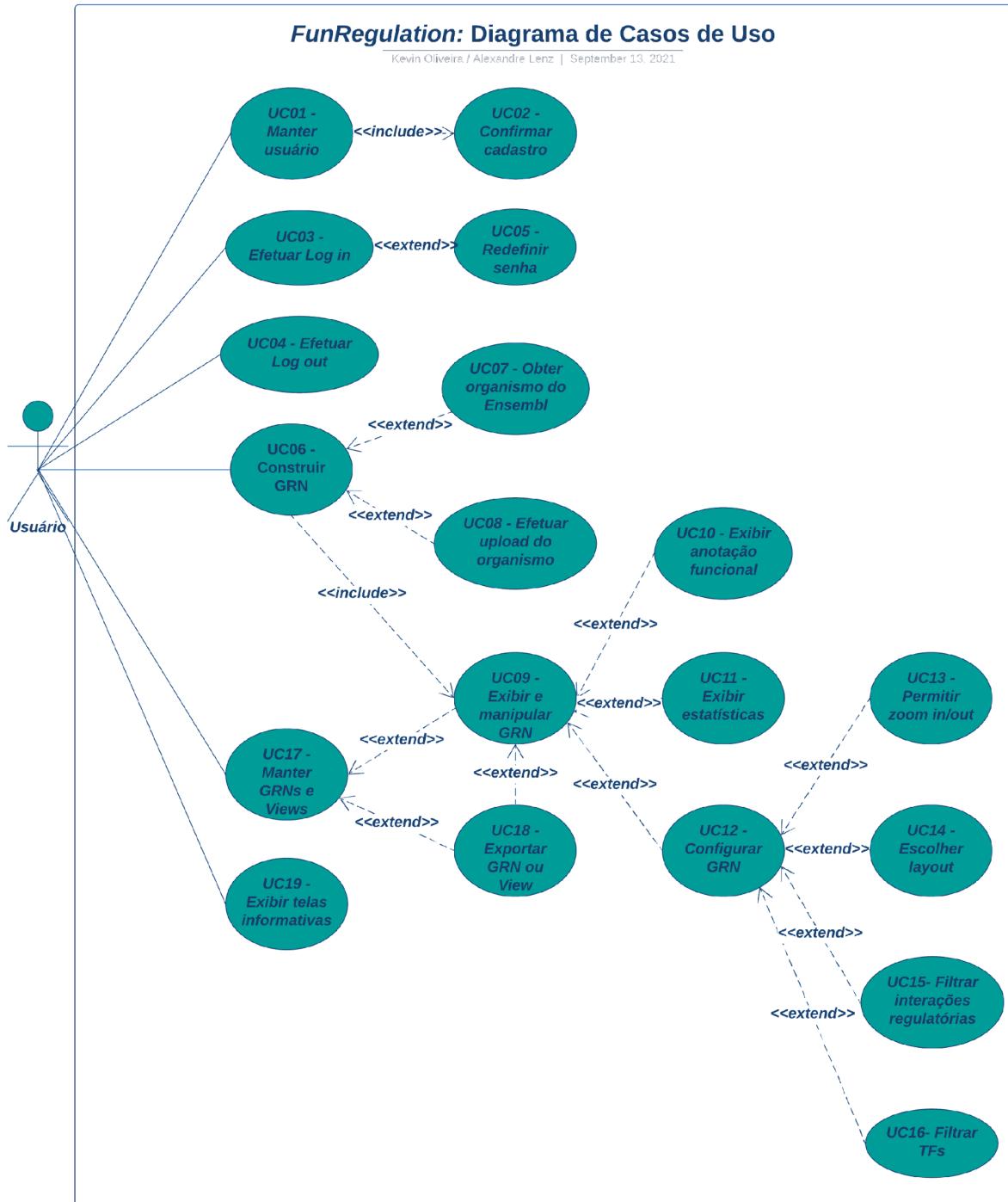
Sumário

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

	<p>2. O usuário corrige os campos de senha; 3. Retorna para o passo 10 do Fluxo normal.</p>
Pós-condições:	Usuário cadastrado no sistema.

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

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

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

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

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

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

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

	<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> <ul style="list-style-type: none"> 4.2.1. Executar os passos 5.1.1 a 5.1.3; 4.2.2. Extrair os promotores 1000bp upstream de cada TG do genoma do organismo alvo; 4.2.3. Obter as matrizes PWM do banco de dados CIS-BP; 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; 4.2.5. Compilar os resultados do RSAT; <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>
Fluxos alternativos:	As alternativas foram descritas no fluxo normal para facilitar a compreensão.
Fluxos de exceção:	<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"> 1. O sistema deve mostrar uma mensagem ao usuário informando o erro que ocorreu; 2. O sistema deve armazenar registro de Log e enviar e-mail, conforme RNF04;
Pós-condições:	A GRN é construída e armazenada.

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

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

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

	6. O sistema retorna ao passo 3 do Fluxo normal do UC06.
Fluxos alternativos:	Não se aplica.
Fluxos de exceção:	<p>2e. O gênero informado pelo usuário não atende à RN05.</p> <p>1. O sistema mostra mensagem indicando que a construção da GRN somente deve ser efetuada para organismos das 4 ordens: <i>Eurotiales</i>, <i>Sordariales</i>, <i>Saccharomycetales</i> e <i>Hypocreales</i>.</p> <p>5e. O usuário não fez upload dos arquivos obrigatórios.</p> <p>1. O sistema mostra mensagem indicando quais são os arquivos obrigatórios.</p> <p>5e. A validação dos arquivos identificou incompatibilidade nos formatos dos arquivos.</p> <p>1. O sistema mostra mensagem indicando quais arquivos não estão no formato correto.</p>
Pós-condições:	Os dados genômicos do organismo são carregados no sistema para construção da GRN.

UC09 - Exibir e manipular GRN		
Subject:	FunRegulation	
Atores:	Usuário	
Objetivo:	Essa funcionalidade compreende a exibição gráfica da GRN e a manipular da mesma a partir de uma série de configurações.	
Modificado por:	Alexandre Lenz	
Versão: 2.0	Status: Aprovado	Prioridade: Alta
Fluxos de eventos		
Pré-condições:	O usuário precisa ter executado o UC06 ou UC17.	
Fluxo normal:	<p>1. O sistema exibe tela com a representação gráfica da GRN, conforme RN06 e RN07;</p> <p>2. O sistema permite uma série de ações a partir da tela de visualização da GRN:</p> <p>2.1. O usuário seleciona um nó da GRN para exibir anotação funcional e o sistema executa o UC10;</p>	

	<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>
Fluxos alternativos:	Não se aplica
Fluxos de exceção:	Não se aplica.
Pós-condições:	GRN exibida na tela.

UC10 - Exibir anotação funcional		
Subject:	FunRegulation	
Atores:	Usuário	
Objetivo:	Mostrar anotação funcional referente a cada gene da GRN.	
Modificado por:	Alexandre Lenz	
Versão: 2.0	Status: Aprovado	Prioridade: Média
Fluxos de eventos		
Pré-condições:	O usuário deve estar executando o caso UC09.	
Fluxo normal:	<ol style="list-style-type: none"> O usuário clica com botão esquerdo em um gene da GRN; 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. O sistema exibe as informações de anotação funcional do gene selecionado em uma janela modal; O usuário fecha a janela modal; O caso de uso é encerrado. 	
Fluxos alternativos:	Não se aplica.	

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

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

UC12 - Configurar GRN	
Subject:	FunRegulation

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

UC13 - Permitir zoom in/out	
Subject:	FunRegulation

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

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

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

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

Pré-condições:	O usuário deve estar executando o caso UC12.
Fluxo normal:	<ol style="list-style-type: none"> 1. 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"> 1.1. Somente Ortologia (Off) - Protein-DNA e Ortologia (On); 1.2. A opção <i>default</i> é Protein-DNA e Ortologia (On); 2. O usuário altera o filtro das interações regulatórias; 3. O sistema aplica o filtro selecionado na View, conforme RN07; 4. O sistema retorna ao passo 2 do Fluxo normal do UC12.
Fluxos alternativos:	Não se aplica.
Fluxos de exceção:	<p>1e. Caso no passo 2.1 do Fluxo normal do UC06 tenha sido escolhida a opção “Somente ortologia de organismo modelo”.</p> <ol style="list-style-type: none"> 1. O botão On-Off fica desabilitado na opção Somente Ortologia (Off).
Pós-condições:	As interações regulatórias mostradas na GRN são filtradas.

UC16 - Filtrar TFs		
Subject:	FunRegulation	
Atores:	Usuário	
Objetivo:	Filtrar os fatores de transcrição que devem ser mostradas na GRN.	
Modificado por:	Alexandre Lenz	
Versão: 1.0	Status: Aprovado	Prioridade: Média
Fluxos de eventos		
Pré-condições:	O usuário deve estar executando o caso UC12.	
Fluxo normal:	<ol style="list-style-type: none"> 1. 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"> 1.1. O usuário pode informar até 10 TFs (locus tag e/ou gene symbol) separadas por “;” 1.2. O campo de texto deve ser mostrado com um exemplo de preenchimento (FOXG_01234;CreA); 	

	<p>2. O usuário informa os TFs e aciona a opção Filtrar TFs;</p> <p>3. O sistema aplica o filtro de TFs na View, são mostrados apenas os TFs informados e os TGs regulados pelos TFs informados;</p> <p>4. O sistema retorna ao passo 2 do Fluxo normal do UC12.</p>
Fluxos alternativos:	Não se aplica.
Fluxos de exceção:	<p>1e. Caso o usuário informe um locus tag ou gene symbol que não é um TF ou que não tenha sido encontrado na GRN;</p> <p>1. O sistema mostra mensagem de erro indicando que o(s) TF(s) informado(s) não foi(ram) encontrado(s).</p>
Pós-condições:	Apenas os TFs informados e seus respectivos TGs são mostrados na GRN.

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

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

UC18 - Exportar GRN ou View

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

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

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