Desenvolvimento de uma ferramenta para geração de grafos a partir de vias de ativação de doenças

Henrique Martins Leal[1], Sylvio Andre Garcia Vieira[1]

[1]Sistemas de Informação – Centro Universitário Franciscano

Caixa Postal – 91.501-970 – Santa Maria – RS – Brazil

{henriquemartinsleal,sylviovieira}@gmail.com

Abstract: *With the increasing number of biological researches there is also an increasing need in analysing the information obtained and stored in databases. Therefore, this union between the biology area and the computation area is very important to this task. When refering to proteic bonds, this relation between computing and biology can further contribute in identifying and evaluating proteic groups, that when mutated can activate disease pathways. To acheive this in a computational way, we must use the latest technologies available, with Python being one of the best choices for developing an application that runs on cloud-computing and which can scale under demand and provide reduced costs, reliability, security and good performance.*

Resumo: *Com o constante aumento do número de pesquisas na área de biologia surge a necessidade de analisar essas informações contidas em bases de dados. Sendo assim, a união entre biologia e informática mostra-se importante para a realização dessa tarefa. Tratando especificamente de ligações proteicas a ligação entre ambas as áreas se mostra pertinente, tendo em vista que podem contribuir na avaliação e identificação do grupo de proteínas que, quando sofrem alterações em conjunto, podem vir a desencadear doenças as quais denominamos de vias de ativação de doenças. Para realizar essa tarefa em nível computacional deve-se recorrer as mais atuais tecnologias disponíveis, sendo uma boa escolha de linguagem de programação o Python e como sede para essa aplicação uma base na cloud-computing que possui como características escabilidade sob demanda, redução de custos, confiabilidade, segurança e desempenho.*

1. Introdução

Com o aumento do número de pesquisas na área da biologia e consequentemente com o aumento dos dados coletados nessa área, surge a necessidade de realizar a análise de

informações contidas nos bancos de dados que mapeiam essas informações para que seja possível acessá-las de forma prática e eficaz.

A união entre biologia e informática tem facilitado o desenvolvimento de pesquisas que contam com inúmeros dados a serem processados e analisados. No caso das ligações protéicas, a ligação entre as mesmas se mostra oportuna, de forma que podem contribuir na avaliação e identificação do grupo de proteínas que, quando sofrem alterações conjuntamente desencadeiam doenças.

Para contribuir com a compreensão dessas interações e visualizá-las na internet, definiu-se como objetivos deste trabalho, o desenvolvimento de um software *web*, que através da análise das proteínas que compõem as vias de manutenção do genoma (GMM, do inglês *Genome Mechanism Maintenance*) permita a construção de um diagrama, através de um grafo, das interações físicas e que venha a representar uma rede de doenças humanas. Os nós desse grafo representariam as doenças e as proteínas expressas em cada uma delas. Já as arestas representam as interações entre elas.

2. Levantamento Bibliográfico

Nesse tópico são apresentados os conceitos relevantes para o entendimento desse trabalho de pesquisa. Serão mostrados conceitos-chave sobre bioinformática que serão utilizados na realização desse trabalho, bem como conceitos sobre computação incluindo bancos de dados e computação nas nuvens.

2.1 Proteínas

Proteínas são macromoléculas, ou seja moléculas de grande dimensão, que são constituídas basicamente por aminoácidos os quais o nosso corpo e as suas células necessitam para funcionar corretamente. Nossas estruturas corporais, funções, a regulação de células, tecidos e órgãos do corpo não podem existir sem proteínas.

Cada célula do corpo humano contém proteínas e elas são uma parte importante da pele, dos músculos, dos órgãos e glândulas. São encontradas em todos os fluidos corporais, exceto na bílis e na urina. Essas moléculas também são necessárias para uma dieta equilibrada, pois ajudam as células de restituição corporal. Além disso auxiliam no crescimento infantil, na adolescência e durante período gestacional. [Medline 2011].

2.2 Computação em nuvem

A computação em nuvem é um modelo que permite o acesso ubíquo e sob demanda de rede a um conjunto compartilhado de recursos computacionais configuráveis (por exemplo, redes, servidores, armazenamento, aplicações e serviços) e que podem ser rapidamente provisionados e liberados com um baixo esforço de gerenciamento ou que possua o mínimo de interação com o provedor de serviços [Mell et al. 2011].

Com essas características inerentes, como escalabilidade sob demanda, redução de custos, confiabilidade, segurança e desempenho, essa plataforma foi escolhida para o desenvolvimento da aplicação. Pois permite o processamento e o armazenamento de dados de acordo com a demanda fornecida, ficando disponível para acesso à *Internet*.

2.3 Base de dados

Para Date (2004) um sistema de banco de dados é basicamente um sistema computadorizado de manutenção de registros, é um repositório para uma coleção de arquivos de dados armazenados de forma digital.

A quantidade de bancos de dados públicos cresce de forma exponencial, e eles são, em geral, formados por uma coleção de informações que sejam relevantes a um determinado assunto como genes, proteínas e suas interações. Para servir de base para esse trabalho foi escolhida a ontologia Ontocancro[1], desenvolvido e mantida por uma parceria entre a Universidade Federal de Santa Maria (UFSM) e o Centro Universitário Franciscano (UNIFRA).

A Ontocancro compila as vias de genes envolvidos no GMM, compreendendo mecanismos de reparo do DNA, ciclo celular, apoptose e senescência. Esse tipo de informação está divulgada na literatura científica e em ontologias. Devido à grande heterogeneidade dos dados destas ontologias e a ausência de uma ferramenta especializada no GMM é difícil construir modelos que envolvem estas vias. Essa ontologia visa facilitar a modelação GMM fornecendo uma fonte integrada de informação [Ontocancro 2012].

[1] Localizada no seguinte endereço da internet http://ontocancro.inf.ufsm.br/

2.4 Metodologia de desenvolvimento

A metodologia cascata possui fases de desenvolvimento que prosseguem em uma ordem estrita, sem qualquer sobreposição ou passos iterativos [Pressman 2006]. Esse método foi escolhido pois os requisitos são conhecidos, a tecnologia usada é acessível e os recursos para desenvolvimento estão disponíveis. O ciclo de desenvolvimento do *software* proposto está divido nas seguintes etapas conforme a Figura 1.

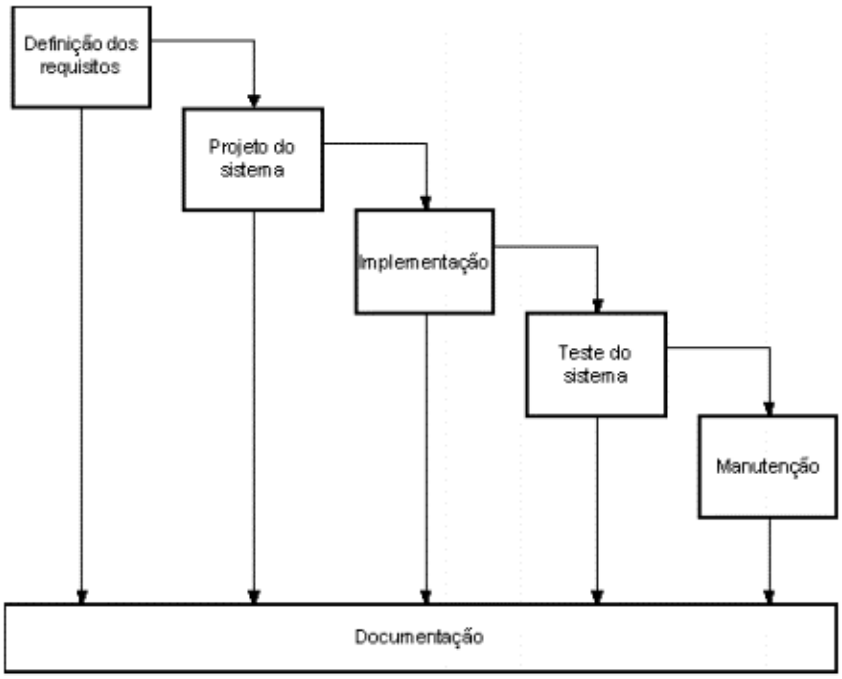

Figura 1. Representação do Ciclo de Vida em Cascata. [Pressman 2006]

3. Ferramentas de apoio

Com base na dificuldade de encontrar ferramentas computacionais simples e capazes de realizar uma filtragem nos dados biológicos da base Ontocancro, optou-se pela criação de um software capaz de suprir essa demanda para visualizar e integrar os dados. Visando a criação de um ambiente científico sustentável, optou-se pela escolha de tecnologias livres e abertas, ou que possuam licença para uso acadêmico.

Dentre as linguagens de programação disponíveis, uma das que tem se destacado na comunidade científica é a linguagem Python. Existem várias bibliotecas para computação científica que são amplamente adotadas no meio acadêmico, entre elas podemos citar as bibliotecas *BioPython*, *NumPy* e *SciPy*. A vasta gama de bibliotecas

para este tema e a facilidade de expressão presente na linguagem torna ela atrativa tanto para pesquisadores e universitários, quanto para empresas que desejam fazer uso comercial.

Levando em consideração que deseja-se desenvolver uma aplicação que permita o acesso através da *Internet* e seja capaz de ser executada sem dificuldades em um ambiente de computação em nuvem, optou-se por escolher o *framework Django*. Ele é uma *framework* voltado para o desenvolvimento *web*, utilizando um padrão de desenvolvimento em três camadas, utilizando o padrão MVT (em inglês Model, View e Template) [MOORE et al, 2007].

No atual cenário de desenvolvimento web é possível criar aplicações Django rapidamente e colocá-las em produção nos mais diferentes provedores de plataforma como serviço (Google AppEngine, Amazon EC2, Heroku) e isso vai de encontro com a necessidade do trabalho.

Já para a parte do armazenamento dos dados que a aplicação será responsável, foi escolhido o SGBD MySQL, pois além de robusto e altamente escalável foi desenvolvido para trabalhar com dados relacionais com foco em web e velocidade, o que converge com os dados que serão utilizados nesse trabalho.

Ao contrário de bancos de dados relacionais, bancos de grafos possuem ligações entre os nós de um grafo, permitindo um alto nível de clusterização [ROBINSON et al., 2013]. Um banco de dados de grafos seria uma outra opção viável para o desenvolvimento desse trabalho. Durante a etapa de pesquisa, o SGBD Neo4J mostrou-se bastante interessante para utilização, porém, quando efetuados alguns testes verificou-se que a biblioteca de integração com o mapeador de objeto relacional ou ORM (do inglês *Object-relational mapping*) do *framework* escolhido está defasada além da integração ser um pouco difícil.

Outro motivo pelo qual esse conjunto de ferramentas foi escolhido é que além de isoladamente possuírem um bom desempenho e também serem bem documentadas, possuem uma boa integração. Possuem também boas bibliotecas para integração como por exemplo, a *ORM* padrão do *framework* para utilização do banco de dados *MySQL*.

3.1 Metodologia

A metodologia escolhida para o desenvolvimento desse trabalho foi a metodologia cascata, pois o desenvolvimento do software possui recursos bem definidos e etapas claras. Sendo assim um fluxo linear e sequencial de atividades.

3.1.1. Desenvolvimento dos requisitos

A análise de requisitos é a primeira fase de desenvolvimento de software. Ela abrange a atividade onde se deve estabelecer os requisitos do produto a ser desenvolvido. Os requisitos devem ser definidos de forma apropriada para que sejam aproveitados na próxima etapa. Essa etapa também inclui a documentação e o estudo da viabilidade do projeto com o fim de determinar o processo de início de desenvolvimento do sistema.

A modelagem de um sistema ajuda o analista a entender todas as suas funcionalidades de maneira mais concreta, já que é através dela que realizamos as descrições abstratas referentes aos requisitos que foram analisados [GRADY et al., 2006]. A UML é uma linguagem para modelagem de sistemas de *software* intensivos, e que foi utilizada para a melhor visualização do sistema desenvolvido através de um diagrama de caso de uso, conforme detalhado na Figura 2.

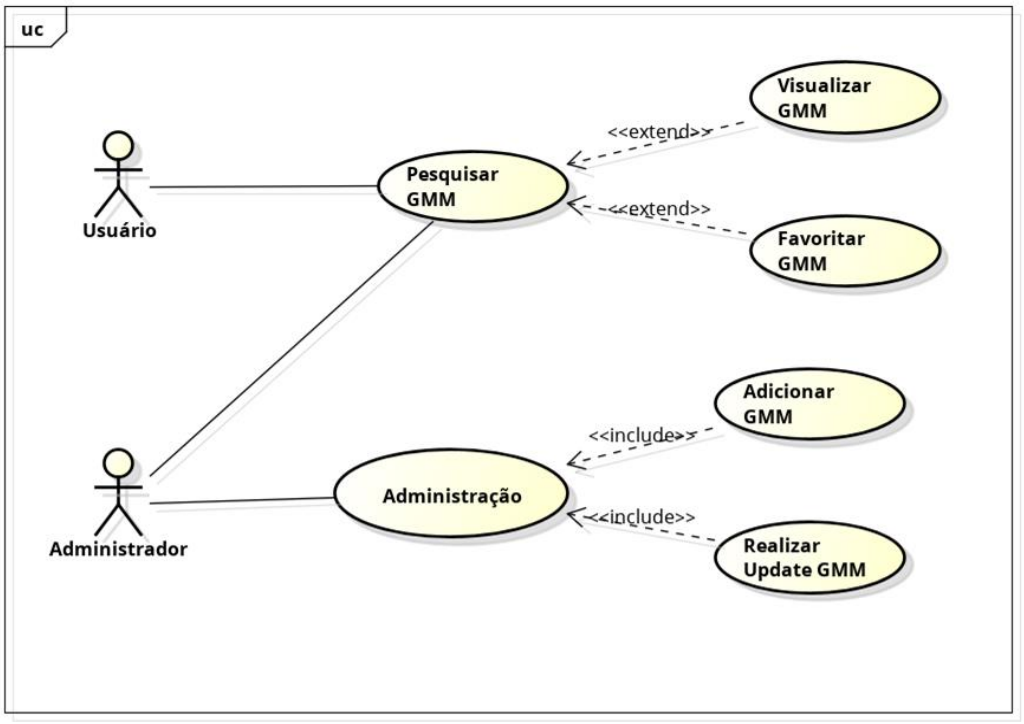

Figura 3. Diagrama de Caso de Uso

3.1.2. Projeto do sistema

A modelagem de sistemas de software é uma técnica da Engenharia de *Software* aprovada e bem aceita na comunidade de desenvolvimento de sistemas na qual os modelos são construídos para auxiliar a construção de um novo sistema ou no processo de aperfeiçoamento de um já existente e utilizado. Nessa etapa são definidos os aspectos físicos e tecnológicos concentrando-se em alguns atributos do sistema: a estrutura de dados, a arquitetura do *software*, a modelagem do sistema definição da interface gráfica, definição do banco de dados e outras considerações.

O Diagrama de Classes, apresentado na Figura 4, tem como finalidade mostrar a representação das relações e estrutura das classes que servem como modelo para os objetos.

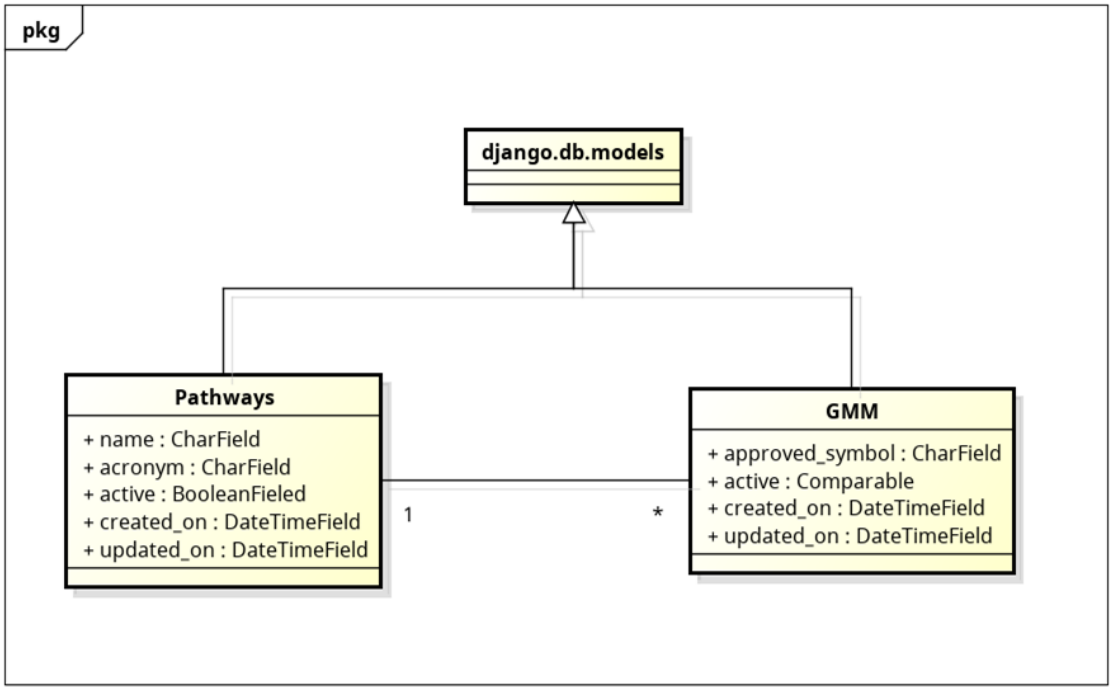

Figura 4. Diagrama de Classes

A partir desses diagramas e requisitos foi definido um projeto de tela para o sistema que pode ser verificado na Figura 5. Nele são apresentadas duas áreas distintas: a área 1 é onde serão listadas as vias de manutenção disponíveis para análise e a área 2 o local onde essas vias poderão ser analisadas na forma de grafos, podendo ser manipuláveis e permitindo uma melhor visualização.

3.1.3 Ambientes de desenvolvimento

Para o desenvolvimento do trabalho, foi utilizado um computador com o sistema operacional *Arch Linux*, com a linguagem de programação Python na sua versão 3.4 instalado e o servidor de banco de dados *MySQL*. Como controlador de versão foi utilizado o *Git* e o código fonte encontra-se atualmente no *GitHub* através do endereço: http://github.com/hmleal/pitagoras/ e encontra-se sobre a licença *MIT*, ou seja, uma licença que permite a reutilização do software e o licenciamento em programas livres ou proprietários.

Já para o ambiente de produção o sistema continua rodando sobre um sistema Linux sobre uma plataforma de serviço chamada *Heroku*, que permite a hospedagem de aplicações e o gerenciamento dos recursos usados pela aplicação. Conta com um sistema de atualizações automatizadas que utiliza as ferramentas desse provedor de serviço.

Para o desenvolvimento das telas do sistema foi adotado um framework chamado *Twitter bootstrap* que possui um conjunto de *widgets* personalizáveis para a construção de páginas web. Sendo a Figura 05 a tela inicial do sistema.

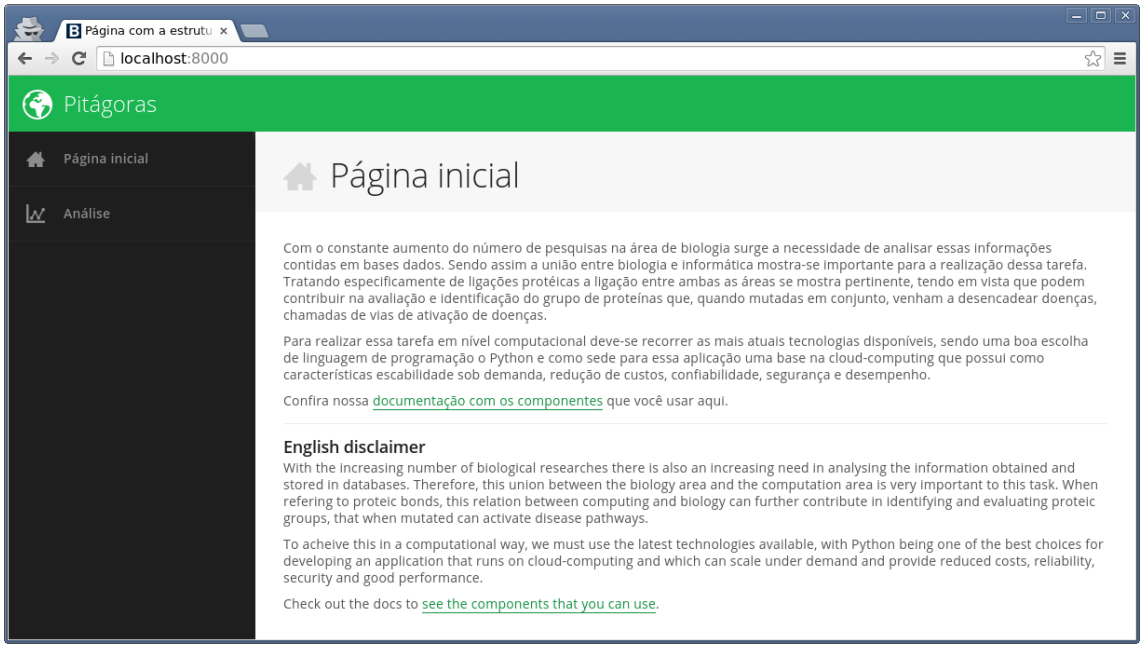

Figura 5. Tela inicial do sistema.

A alimentação dos dados no sistema foi feita utilizando o app django-admin que vem junto com o framework. Já que ela possui excelente integração e possui todo um

sistema de acesso e controle aos dados imputados no sistema. Na Figura 06 é possível ver uma dessas telas de cadastro.

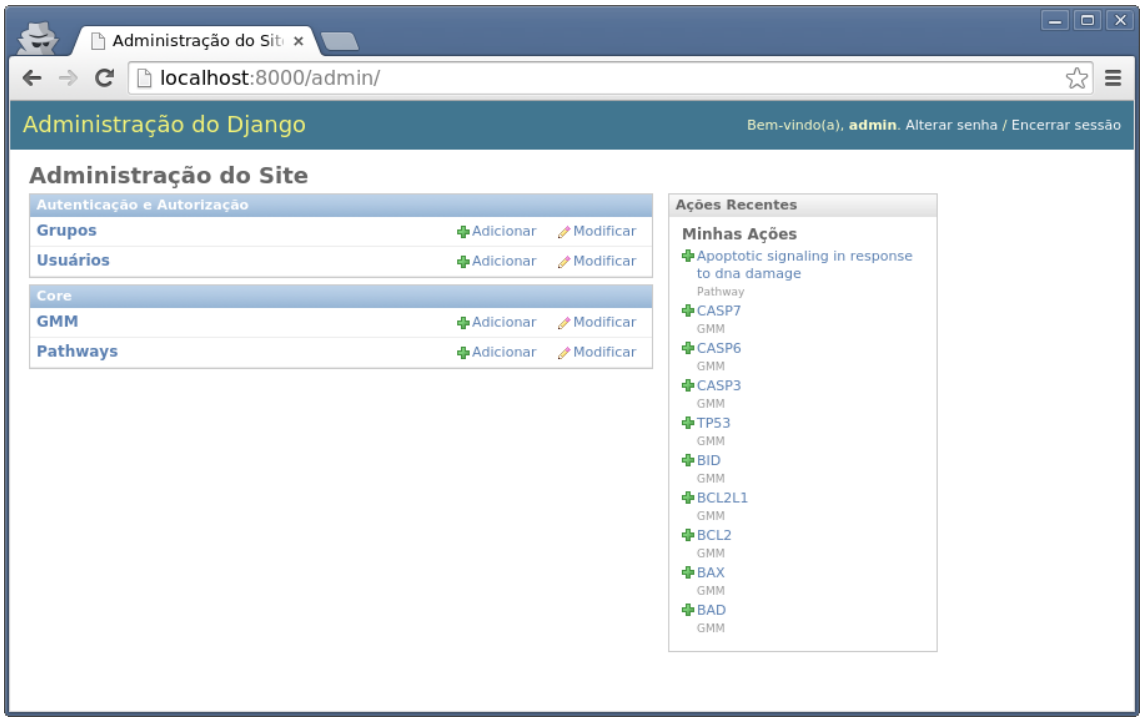

Figura 6. Painel administrativo do sistema.

3.2 Dados

A Ontocancro disponibiliza os dados em arquivos de texto simples, após o download e conversão dos dados conforme o diagrama de classes mostrados na Figura 04, os dados importados da ontologia ficaram no formato mostrado na Figura 7. Respeitando a estrutura relacional de banco de dados.

id	approved_symbol	active	created_on	updated_on
1	AKT1	1	2014-10-27 01:00:36.441195	2014-10-27 01:00:36.441268
2	ATM	1	2014-10-27 01:00:40.467010	2014-10-27 01:00:40.467078
3	CYCS	1	2014-10-27 01:00:47.889241	2014-10-27 01:00:47.889299
4	BAD	1	2014-10-27 01:00:53.755059	2014-10-27 01:00:53.755160
5	BAX	1	2014-10-27 01:00:58.286838	2014-10-27 01:00:58.286933
6	BCL2	1	2014-10-27 01:01:05.095081	2014-10-27 01:01:05.095156
7	BCL2L1	1	2014-10-27 01:01:10.132956	2014-10-27 01:01:10.133048

Figura 7. Exemplo da estrutura de dados do sistema.

4. Resultados

Este capítulo tem como objetivo apresentar e discutir as principais observações feitas durante o decorrer do trabalho desta pesquisa, ao mesmo tempo em que busca analisar os resultados obtidos durante o desenvolvimento do software proposto.

4.1 Grafo

Com os dados oriundos da Ontocancro, foi possível aplicar a ferramenta desenvolvida para a geração dos grafos. Após a organização dos dados na estrutura relacional proposta chegamos à representações das integrações da vias, como pode ser observado na Figura 8.

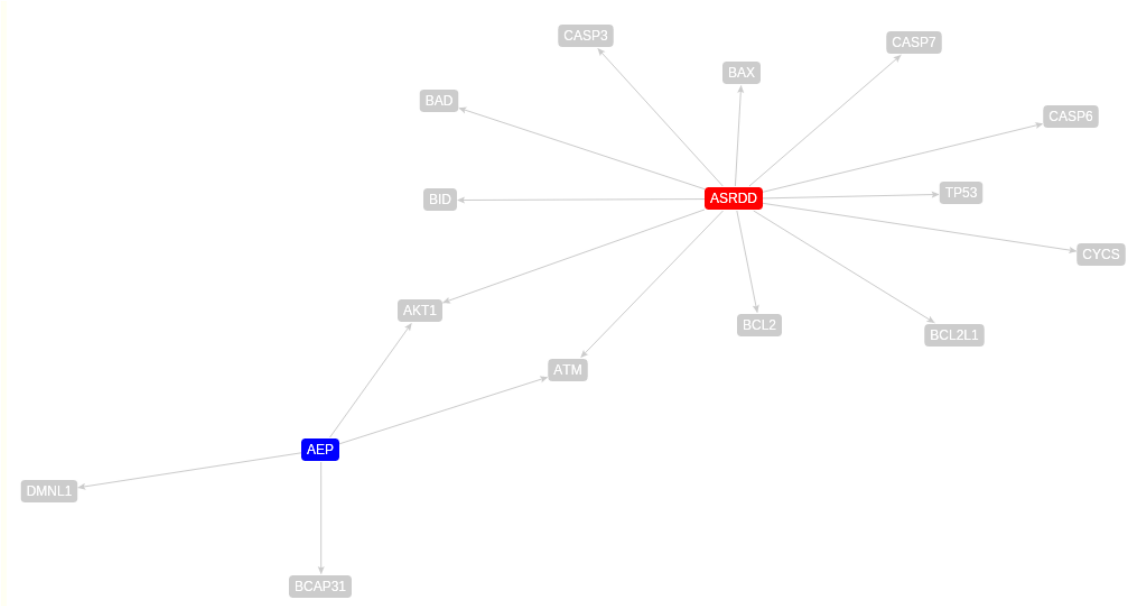

Figura 8. Exemplo de grafo gerado pelo sistema mostrando apenas 2 vias de ativação.

5. Conclusões

A linguagem Python mostrou-se eficaz e bastante rápida na montagem dos diagramas propostos, sendo integrável ao banco de dados e a exibição destes na web.

O *MySQL* mostrou-se bastante ágil no fornecimento dos dados para o montagem dos Grafos, o que já era esperado por ser um banco bem conhecido pela sua rapidez e confiabilidade e pela boa integração com o ambiente proposto ao desenvolvimento. Outro banco de dados de grafos que se mostrou bastante promissor mas que não chegou a ser implementado foi o *OrientDB* que possui uma excelente ferramenta de visualização dos dados que nele inseridos, mas como não possui uma biblioteca de integração com *python* até momento foi descartado.

O desenvolvimento do *front-end* com utilização do *Bootstrap* do *Twitter* permitiu que as telas do sistema se adeque aos mais variados dispositivos. Já para uma boa visualização dos gráficos do sistema é necessário uma tela de no mínimo 800 x 600 que é geralmente disponível em qualquer *tablet*.

As propriedades biológicas das proteínas encontradas, não foram pesquisadas por não fazerem parte do escopo deste trabalho.

A solução indicada no projeto teve bons resultados, aliado às tecnologias adotadas. A sequência de softwares foi utilizada a contento, sendo o único contratempo a escolha do banco de dados *Neo4j* como base de dados adotada.

A biblioteca *Neo4Django* responsável pela integração entre o *framework* escolhido e o banco de dados *Neo4J* mostrou-se um pouco defasada sendo sua última atualização há mais de um ano. Sendo assim foi optado a escolha do banco de dados *MySQL* para o desenvolvimento do sistema.

Em continuidade a este trabalho, sugere-se o desenvolvimento de uma integração do Python com o banco de dados *OrientDB*, o que pode contribuir bastante com as pesquisas biológicas e melhorar a visualização dos dados contidos na base de dados desse trabalho e em muitas outras disponíveis nos bancos de dados biológicos na internet.

5. Referências Bibliográficas

DATE, C.J., Introdução a Sistemas de Banco de Dados, 8a ed., Campus, 2004.

HEIDEN M. G. V. , CANTLEY L. C., THOMPSON C. B., Understanding the Warburg effect: the metabolic requirements of cell proliferation. Science 324: 1029–1033, 2009.

MEDLINE MEDICAL ENCYCLOPEDIA, Protein in diet. 2011. Disponível em: http://www.nlm.nih.gov/medlineplus/ency/article/002467.htm, acessado em 23 de maio de 2014.

MELL, P. , GRANCE, T., The NIST Definition of Cloud Computing, 2011. Disponível em: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf, acessado em 23 de maio de 2014.

MOORE, D.; BUDD R. e WRIGHT W. “Professional Python® Frameworks Web 2.0 Programming with Django® and TurboGears™”. Wiley Publishing, Inc, 2007.

ONTOCRANCRO 2.0. Disponível em: http://ontocancro.inf.ufsm.br/, acessado em 23 de maio de 2014.

PRESSMAN, R. S. Engenharia de Software. 6º ed. Rio de Janeiro: McGraw-Hill, 2006.

ROBINSON, I. WEBBER, J. EIFREM, E. Graph Databases, 1a ed., O'Reilly Media, Inc, 2013.

TECHTERMS, The Tech Terms Computer Dictionary, Disponível em: http://www.techterms.com/definition/database acessado em 29 de maio de 2014.