

ANDERSON ANDREATTA

**BAK4BIO FRAMEWORK - BRAZILIAN
ARMY KNIFE FOR BIOINFORMATICS**

CURITIBA

2013

ANDERSON ANDREATTA

**BAK4BIO FRAMEWORK - BRAZILIAN
ARMY KNIFE FOR BIOINFORMATICS**

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

Orientador: Prof. Dr. Alessandro Brawerman

CURITIBA

2013

Termo de Aprovação

ANDERSON ANDREATTA

BAK4BIO FRAMEWORK - BRAZILIAN

ARMY KNIFE FOR BIOINFORMATICS

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

Prof. Dr. Alessandro Brawerman
Universidade Federal do Paraná

Prof^a Dr^a Jeroniza Nunes Marchaukoski
Universidade Federal do Paraná

Prof. Dr. Pedro Luis Kantek G. Navarro
Universidade Positivo

Dedicatória

Aos meu pais, Macir e Cleotildes.

A minha esposa, Tamires.

As minhas irmãs, Fabiana, Danielle e Laize.

Aos meus amigos Renan, Ericson e Daniel.

Agradecimentos

A todos que, direta ou indiretamente, contribuíram para a realização e divulgação deste trabalho.

Epígrafe

O que sabemos é uma gota,
o que ignoramos é um oceano.

Isaac Newton

Sumário

Lista de Figuras	ix
Lista de Tabelas	xi
Lista de Siglas	xii
Resumo	xiii
Abstract	xiv
1 INTRODUÇÃO	1
1.1 Objetivos.....	2
1.1.1 Objetivo Geral	2
1.1.2 Objetivo Específicos.....	2
1.2 Justificativa e relevância do trabalho	3
1.3 Organização da Dissertação	4
2 REVISÃO DA LITERATURA	5
2.1 Aplicativos Móveis.....	5
2.1.1 Evolução dos Celulares	5
2.1.2 Android	6
2.2 Integração de Sistemas via Web	8
2.2.1 SOAP - <i>Simple Object Access Protocol</i>	9
2.2.2 REST - <i>Representational State Transfer</i>	10
2.2.3 JSON - <i>JavaScript Object Notation</i>	12

2.3	Banco de Dados Biológicos	12
2.4	Ambiente Computacional em Cluster	14
2.5	Ruby on Rails	16
2.6	Controle de versão - Git	16
2.7	Computação em nuvem	17
2.8	Padrões de Projeto	18
2.8.1	MVC	19
2.9	Trabalhos Relacionados	20
2.9.1	Serviços Web	20
2.9.1.1	TogoWS	20
2.9.1.2	<i>A Multi-Protocol Bioinformatics Web Services e HitKeeper</i>	21
2.9.1.3	WS-BioZard	22
2.9.2	Dispositivos Móveis	22
2.9.2.1	BioWAP	22
2.9.2.2	<i>SimGene e SimAlign</i>	23
2.9.2.3	Aplicativo FlyExpress	24
2.9.2.4	<i>Hematopoietic Expression Viewer</i>	24
2.9.3	Supervisão de <i>Workflows</i>	24
2.9.3.1	SciCumulus	24
2.9.3.2	SciLightning	25
2.9.3.3	Submissão e Monitoração de Tarefas em Grid	25
3	MATERIAIS E MÉTODOS	26
3.1	Visão Geral	26
3.2	BAK4BIO Server	27
3.2.1	Autenticação e autorização	28
3.2.2	Proxy	30

3.3	BAK4BIO Box	32
3.4	BAK4BIO Cluster	34
3.5	BAK4BIO Droid	35
3.5.1	Modelagem	35
3.5.2	Acesso a base de dados biológicas	36
3.5.3	Execução de BLAST	38
3.5.4	Segurança	39
3.5.5	Distribuição	40
3.6	BAK4BIO Web	40
3.6.1	Funcionalidades	40
3.7	Documentação	41
4	EXPERIMENTOS PRÁTICOS E RESULTADOS	42
4.1	BAK4BIO Droid	42
4.1.1	Usuários	42
4.1.2	Distribuição geográfica	43
4.1.3	Versões do Android	43
4.2	Estudo de caso	45
4.2.1	Acesso a base de dados biológicas	45
4.2.2	Execução de BLAST	47
5	CONCLUSÃO	50
6	TRABALHOS FUTUROS	52
6.1	Geral	52
6.2	BAK4BIO Droid	52
6.3	BAK4BIO Web	52
6.4	BAK4BIO Box	52

6.5	BAK4BIO Server	53
6.6	BAK4BIO Cluster	53
	Referências	54
	Anexo A – BAK4BIO Droid: Caso de Uso	58
	Anexo B – BAK4BIO Web: Caso de Uso.....	59
	Anexo C – BAK4BIO Server: Caso de Uso.....	60
	Anexo D – BAK4BIO Box: Caso de Uso	61
	Anexo E – BAK4BIO Cluster: Caso de Uso.....	62
	Anexo F – BAK4BIO Droid: Diagrama de Classes	63
	Anexo G – BAK4BIO Server: Diagrama de Classes	64
	Anexo H – BAK4BIO Cluster: Diagrama de Classes	65
	Anexo I – BAK4BIO: Modelagem Banco de Dados	66

Lista de Figuras

Figura 1	Funcionamento de um <i>Webserie</i>	8
Figura 2	Funcionamento de um <i>Webserie</i> SOAP.	10
Figura 3	Arquitetura de um <i>Webserie</i> RESTful.	11
Figura 4	Exemplo REST com JSON.	13
Figura 5	Estrutura do INSDC.	14
Figura 6	Exemplo da arquitetura presente em computação nas nuvens.	18
Figura 7	Padrão de projetos MVC	19
Figura 8	Arquitetura do projeto TogoWS.	21
Figura 9	Fluxo de requisições no BAK4BIO.	27
Figura 10	Arquitetura do Framework BAK4BIO.	28
Figura 11	Visão geral do processo de autenticação.	29
Figura 12	Visão geral do processo de autorização.	30
Figura 13	Comunicação entre os módulos.	31

Figura 14	Procedimento de armazenamento do módulo Box.	32
Figura 15	Interface de acesso ao módulo Web.	33
Figura 16	Interface de listagem aos arquivos.	33
Figura 17	Fluxo de atividade do BAK4BIO Cluster.	34
Figura 18	Estrutura inicial de um aplicativo Android.	36
Figura 19	Integração com o TogoWS.	37
Figura 20	Demonstração do fluxo de trabalho.	37
Figura 21	Interfaces para a realização do BLAST.	38
Figura 22	Diferentes áreas de acesso do BAK4BIO Droid.	39
Figura 23	BAK4BIO Droid: Downloads totais.	43
Figura 24	BAK4BIO Droid: Distribuição geográfica.	43
Figura 25	Distribuição das versões Android	44
Figura 26	BAK4BIO Droid: Versões do Android	44

Lista de Tabelas

Tabela 1	Seis Maiores Sistemas Operacionais Móveis.	7
Tabela 2	Parâmetros BLAST	39
Tabela 3	Performance integração TogoWS com apenas 1 usuário	46
Tabela 4	Performance integração TogoWS com 10 usuários simultâneos	46
Tabela 5	Performance integração TogoWS com 50 usuários simultâneos	46
Tabela 6	Performance no envió de novas requisições BLAST	48
Tabela 7	Performance na transferência de arquivos de entrada entre os módulos Box e Cluster	48
Tabela 8	Performance no processamento no módulo Cluster	49
Tabela 9	Performance na transferência de arquivos de saída entre os módulos Clus- ter e Box	49

Lista de Siglas

W3C	World Wide Web Consortium
HTTP	Hiper Text Transfer Protocol
XML	eXtensible Markup Language
SOAP	Simple Object Access Protocol
WSDL	Web Services Description Language
REST	Representational State Transfer
JSON	JavaScript Object Notation
MVC	<i>Model View Control</i>
APK	Android Package
SDK	Software Development Kit

Resumo

No desenvolvimento de software para a área de Bioinformática, pesquisadores consideram vários fatores para definir as tecnologias e metodologias utilizadas em um determinado projeto, não sendo raro encontrar soluções com finalidades iguais ou similares utilizando diferentes tecnologias. Com a proposta de fortalecer a comunidade de Bioinformática, o framework BAK4BIO - *Brazilian Army Knife for Bioinformatics* é um conjunto de ferramentas para a construção de aplicativos para a plataforma Web e Mobile. É composto por uma série de subsistemas, os quais possibilitam ao usuário final buscar informações nos principais bancos de dados biológicos; iniciar e supervisionar processos de Bioinformática em ambiente computacional de alto desempenho - BAK4BIO Cluster; e gerenciar arquivos relacionados a entrada e saída de processos nas nuvens - BAK4BIO BOX. Tudo isto provendo a mobilidade necessária para seus usuários, ao permitir com que todos os processos sejam iniciados a partir de um aplicativo desenvolvido para Android - o BAK4BIO Droid. Existe também módulo BAK4BIO Server, responsável por centralizar o recebimento de requisições do módulo Droid e encaminhar para o destino correto, atuando como um proxy. Finalmente, o BAK4BIO Web, criado para interagir com as soluções do framework via Web. A carência de frameworks para o desenvolvimento de ferramentas em Bioinformática encontrada na literatura demonstra que, a existência do BAK4BIO contribui para a integração entre diferentes trabalhos científicos e o coloca em destaque por seu pionerismo em âmbito nacional na área de mobilidade. Com destaque ao módulo BAK4BIO Droid, disponível na Google PlayStore e utilizado pela comunidade científica em vários países desde o seu lançamento.

Palavras-chave: Framework; Dispositivos Móveis; Integração de sistemas; Acesso a bases biológicas; Ferramentas Biológicas.

Abstract

Software development for Bioinformatics, researchers consider several factors to define the technologies and methods used in a given project, it isn't uncommon find solutions with similar purposes using different technologies. With the proposal to strengthen the Bioinformatic's community, the framework BAK4BIO - *Brazilian Army Knife for Bioinformatics* is a set of tools for building applications for the Web and Mobile platforms. It consists of a number of subsystems, which enable the user to retrieve information on major biological databases; initiate and supervise Bioinformatics processes in high-performance computing environment - BAK4BIO Cluster, and manage related to input and output processes in clouds - BAK4BIO BOX. All this providing the necessary mobility to users by allowing that all processes start from an application developed for Android - BAK4BIO Droid. There is also BAK4BIO Server module, responsible for centralizing the receipt of requests from the Droid module and forward to the correct destination, acting as a proxy. Finally, the Web BAK4BIO designed to interact with the framework via Web solutions The lack of frameworks for the development of Bioinformatics tools found in the literature shows that the existence of BAK4BIO contributes to the integration of different scientific studies and highlights in for his pioneering nationwide in the area of mobility. Highlighting the BAK4BIO Droid unit, available in Google PlayStore and used by the scientific community in several countries since its launch.

Key-words: Framework, Mobile Devices, System's Integration, Access to biological bases; Biological Tools.

1 INTRODUÇÃO

O avanço na área de computação se reflete em várias áreas do conhecimento, e isso não é diferente com a Bioinformática. No entanto, o uso destas inovações não ocorre de forma imediata. É o caso dos dispositivos móveis, no qual a Bioinformática atualmente possui um foco de pesquisa um pouco menor, conforme resultados obtidos em pesquisas por artigos científicos e nas principais lojas mundiais de aplicativos para *smartphones* e tablets: Google Play e Apple Store.

Ao explorar soluções móveis e integração de sistemas de informação, este trabalho propõe a construção de um framework para desenvolvimento de softwares aplicados à área de Bioinformática, ao permitir consultas em bases de dados biológicas e gerenciamento de processos em clusters de auto poder de processamento, bem como o armazenamento de documentos científicos de forma centralizada e disponível para qualquer usuário com acesso a Internet.

Nomeado de BAK4BIO - *Brazilian Army Knife For Bioinformatics* ou Canivete Brasileiro para Bioinformática por ser uma ferramenta fundamental para o dia a dia dos profissionais da área e contar com um conjunto de funcionalidades que se complementam e atendem a várias necessidades, da mesma forma que um canivete se destina a fazer.

Composto por uma série de subsistemas que juntos possibilitam ao usuário buscar informações nos principais bancos de dados biológicos; iniciar e supervisionar processos de Bioinformática em um ambiente computacional de cluster de alto desempenho - BAK4BIO Cluster; e gerenciar seus arquivos de dados em um servidor de armazenamento nas nuvens - BAK4BIO BOX. Tudo isto provendo a mobilidade necessária para seus usuários, já que todos os processos são iniciados a partir de um aplicativo móvel desenvolvido para Android - o BAK4BIO Droid. Por fim, para o correto funcionamento dos processos do framework, foi necessário ainda o desenvolvimento dos módulos BAK4BIO Server que centraliza o recebimento das requisições do BAK4BIO Droid e as encaminha para o local adequado, como se fosse um proxy, e o do BAK4BIO Web que é a interface de interação com o

usuário via Web.

1.1 Objetivos

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

1.1.1 Objetivo Geral

O trabalho tem como objetivo elaborar a arquitetura, construção e comprovação da viabilidade sobre o framework BAK4BIO ao construir funcionalidades de consulta as principais bases de dados biológica, possibilitar a execução de BLAST em ambientes de alto poder de processamento, armazenar arquivos de entrada e saída de processos de forma centralizada e na nuvem, bem como explorar a carência de aplicativos para dispositivos móveis em Bioinformática ao realizar o desenvolvimento de um aplicativo com as funcionalidades citadas para o sistema operacional Android. Por fim, contribuir e viabilizar com a integração de outros projetos acadêmicos e científicos da área com o uso efetivo do BAK4BIO.

1.1.2 Objetivo Específicos

Para o desenvolvimento deste framework, foi necessário alcançar um conjunto de objetivos específicos, sendo eles:

- a) Integração aos principais recursos da área: NCBI, EBI, PDBJ, PDB, KEGG e DDJB;
- b) Preparar arquitetura para a possibilidade de integração com outros sistemas da UFPR ao BAK4BIO;
- c) Armazenar arquivos científicos de forma centralizada e na nuvem em um dos servidores da UFPR;
- d) Possibilitar a utilização de arquivos (presentes na nuvem) como entrada e saída de processos;

- e) Desenvolver um aplicativo móvel capaz de gerar requisições aos principais recursos da área: NCBI, EBI, PDBJ, PDB, KEGG e DDJB e para o cluster de processamento da UFPR;
- f) Disponibilizar a utilização do aplicativo para usuários com dispositivos móveis com sistema operacional Google Android;
- g) Desenvolver um módulo gerenciador no servidor Web para centralizar as requisições e as direcionar ao destino correto, sendo este o cluster ou as bases de dados biológicas;
- h) Desenvolver uma interface Web para que o usuário possa utilizar a plataforma BAK4BIO a partir de um navegador Web;
- i) Desenvolver um sistema multi-thread capaz de receber requisições de processos e as executá-las no cluster de processamento de alto desempenho;
- j) Integrar todos estes sistemas através de serviços Web;
- k) Compartilhar os códigos fonte envolvido na solução com a comunidade acadêmica.

1.2 Justificativa e relevância do trabalho

Ao realizar o desenvolvimento de softwares para a área de Bioinformática, os pesquisadores e envolvidos, consideram vários fatores ao definir quais as tecnologias e metodologias a serem utilizadas no projeto, abaixo temos alguns exemplos:

- Para qual plataforma vamos desenvolver?
- Qual o protocolo de comunicação entre sistemas?
- Qual a familiaridade dos envolvidos no projeto sobre determinada tecnologia?
- O quanto é produtivo trabalhar dessas tecnologias?
- A comunidade em Bioinformática já utilizou essa tecnologia? Quais são as referências?

Com as devidas respostas aos questionamentos acima, são construídos softwares para várias finalidades em diferentes tecnologias e metodologias. Até certo momento esse processo é benéfico, porém torna-se um problema ao depararmos com projetos para

finalidades iguais ou semelhantes em tecnologias e/ou metodologias distintas, e possibilitar uma descentralização e enfraquecimento da comunidade ou instituição de ensino.

Com a proposta de fortalecer a comunidade de Bioinformática e principalmente o Programa de Pós Graduação em Bioinformática da UFPR, o BAK4BIO viabiliza a utilização do framework para a construção de soluções para a plataforma Web e Mobile, ao fazer uso das mais recentes tecnologias do mercado e aplicar metodologias e padrões de desenvolvimento para maior produtividade e baixa manutenção, bem como trabalhar com as inovações em dispositivos móveis e firmar o pioneirismo no Brasil ao desenvolver aplicativos para a área.

1.3 Organização da Dissertação

Os próximos capítulos deste documento pretendem apresentar o desenvolvimento deste trabalho. Este documento está dividido da seguinte forma, o Capítulo 2 apresenta uma revisão bibliográfica que visa formar a fundamentação teórica necessária para um melhor entendimento do trabalho. Além disto, este capítulo também apresenta uma coletânea de trabalhos relacionados a diversas tecnologias utilizadas no BAK4BIO, auxiliando ao leitor organizar uma visão do estado da arte.

O Capítulo 3, por sua vez, apresenta a metodologia utilizada para o desenvolvimento do framework BAK4BIO, um conjunto de módulos diversificado que vão desde um aplicativo móvel, passando por um sistema Web, até chegar em um sistema multi-thread em ambiente computacional de cluster. Este capítulo também demonstra como o framework integra os diversos módulos explorando o uso de serviços Web.

A dissertação continua com o Capítulo 4, que apresenta os testes e experimentos práticos executados, bem como uma breve dissertação sobre os resultados obtidos. Por fim, o Capítulo 5 conclui o trabalho e o Capítulo 6 apresenta as necessidades para desenvolvimentos futuros.

2 REVISÃO DA LITERATURA

Este capítulo tem por finalidade apresentar conceitos importantes e fornecer uma fundamentação teórica que servirá como base para um melhor entendimento do trabalho. Além destes, é fornecida também uma visão geral das principais tecnologias utilizadas para o desenvolvimento do BAK4BIO e os principais trabalhos relacionados, seja ao tema abordado ou as tecnologias utilizadas.

Inicialmente, apresenta-se um panorama sobre o desenvolvimento de aplicativos móveis, seguido de uma explicação de integração de sistemas via Web, passando por serviços Web SOAP, REST e codificação de dados JSON. O capítulo continua apresentando os principais bancos de dados biológicos, conceitos de computação em cluster, computação em nuvens e padrões de projetos, especialmente o padrão MVC, muito utilizado no desenvolvimento de aplicativos móveis. Por fim, são apresentados os principais trabalhos correlatos.

2.1 Aplicativos Móveis

Nessa seção são abordados os principais temas referentes a aplicativos móveis, como seu funcionamento e mercado de trabalho, além de um breve histórico.

2.1.1 Evolução dos Celulares

Segundo (FLUHR; NUSSBAUM, 1973), a origem dos dispositivos móveis remonta o ano de 1973, com a realização da primeira chamada telefônica a partir de um dispositivo sem fio para um telefone fixo. O início de sua comercialização aconteceu com a empresa Motorola (MOTOROLA, 2013) no ano de 1983, o Motorola DynaTAC.

Conforme (TEAM, 2010), o hardware de celulares era constituído de um processador mais lento e pouca memória, além de sistemas operacionais fechados (proprietários), os desenvolvedores tinham dificuldades em criar novos aplicativos. No entanto com a

evolução dos celulares, tanto em termos de hardware, quanto sistema operacional, chegou-se ao termo *smartphone*. Segundo definição em (SCOOP, 2008) são celulares inteligentes, com um maior poder de processamento, maior quantidade de memória, sistema operacional multi-tarefa e a possibilidade de desenvolvimento de aplicativos móveis por terceiros.

Um dos maiores impulsos em termos de tecnologia para *smartphones* foi o iPhone, lançado em 2007 pela empresa Apple. Esta criação revolucionou o mercado de dispositivos móveis com um hardware mais avançado e sofisticado (SAGER, 2012). Além disto, a Apple alavancou o mercado de vendas de aplicativos móveis com a AppStore. Após o lançamento do iPhone, outros "gigantes da Informática", como o Google, começaram também a investir neste mercado e desde então a tecnologia dos aparelhos *smartphones* vêm avançando cada vez mais rápido (TEAM, 2010).

Com todo este avanço e baixa nos custos dos aparelhos, o mercado da telefonia móvel tem crescido significativamente nos últimos anos. Conforme pesquisas demonstram, em 2012 as vendas dos *smartphones* cresceram quase 50% em relação ao ano anterior, atingindo a marca de 169 milhões de unidades (RUIC, 2012). A estimativa para 2013 é que o número de unidades vendidas apenas no Brasil chegue a 21,4 milhões (REUTERS, 2012). Resultados como este indicam uma grande tendência de mercado, que promete um ótimo retorno para quem quer investir nele.

Até mesmo na área de Bioinformática, que necessita de uma alta capacidade de processamento e memória para a execução de seus processos, os *smartphones* tem sido utilizados. Não como executores de tarefas, mas sim como visualizadores de informação ou como gerenciadores de processos/*workflows* que executam em clusters de processamento paralelo ou servidores externos, conforme pesquisa realizada pelo termo 'Bioinformatics' nas principais lojas de aplicativos do Mundo, Google Play e Apple Store. Esta, inclusive, é uma das propostas deste trabalho, utilizar o *smartphone*, neste caso com sistema operacional Android, para gerenciar processos ligados a Bioinformática, com a grande vantagem de fornecer mobilidade e informação ao usuário a qualquer momento e em qualquer lugar, desde que haja uma conexão com a Internet.

2.1.2 Android

O Android é uma plataforma de código-fonte aberto introduzido pela Google para o desenvolvimento de aplicativos para dispositivos móveis como *smartphones* ou *tablets*. A plataforma possui uma arquitetura muito flexível, a qual pode integrar aplicações nativas com recém-criadas. A construção do Android foi feita com base no sistema operacional

Linux. Baseado no Kernel 2.6, o sistema tem segurança, gerenciamento de memória e controles de rede e *drivers*. A utilização da linguagem Java é um dos destaques no Android, porém não há uma máquina virtual Java (JVM). Há, na verdade, uma máquina virtual otimizada para utilização em dispositivos móveis, a Dalvik (LECHETA, 2013).

Atualmente, cerca de 1,3 milhão de aparelhos com Android são ativados por dia, segundo Eric Schmidt, presidente do Google (MOTOROLA ON DISPLAY, 2012). O sistema Android está entre os grandes serviços do Google e é o sistema mais usado em celulares, ultrapassando o iOS da Apple. Em 2012, a parcela de mercado do Android foi de 68,4% e a da Apple, de 19,4% (REUTERS, 2012). É possível verificar essa supremacia do Android na Tabela 1:

Sistema Operacional	3º trimestre de 2012		3º trimestre de 2011		Mudança
	Volumes de Remessa	Fatia de Mercado	Volumes de Remessa	Fatia de Mercado	
Android	136.0 mi	75.0%	71.0 mi	57.5%	91.5%
iOS	26.9 mi	14.9%	17.1 mi	13.8%	57.3%
BlackBerry	7.7 mi	4.3%	11.8 mi	9.5%	-34.7%
Symbian	4.1 mi	2.3%	18.1 mi	14.6%	-77.3%
Windows Phone 7 / Windows Mobile	3.6 mi	2.0%	1.5 mi	1.2%	140.0%
Linux	2.8 mi	1.5%	4.1 mi	3.3%	-31.7%
Outros	0.0 mi	0.0%	0.1 mi	0.1%	-100.0%
Total	181.1 mi	100%	123.7 mi	100%	46.4%

Tabela 1: Seis Maiores Sistemas Operacionais Móveis.

Fonte: www.extremetech.com.

Além disso, a pesquisa feita em (RUIC, 2012) revela que aparelhos com Android possuem preços mais atrativos, o que colabora ainda mais para a difusão das tecnologias que esses *smartphones* oferecem. O Google conseguiu com a plataforma Android atender as maiores expectativas de um desenvolvedor que são uma plataforma livre, confiável, robusta, de código aberto e fácil de usar, contando com fabricantes produzindo celulares bonitos, baratos e com diversas funcionalidades. Com estas características o Android não ganhou somente os usuários, mas também os desenvolvedores e milhares de aplicativos em sua loja, a Google PlayStore (PACHECO, 2011).

Com a Google PlayStore, a inserção de produtos no mercado de aplicativos para Android permite que desenvolvedores enviem seus projetos muito facilmente, sem haver um grande número de condições a serem cumpridas e nem preços elevados a serem pagos. É uma grande vantagem que o Android possui em relação as outras grandes empresas que também atuam no desenvolvimento de aplicativos móveis, a Apple e a Microsoft (ANDROID DEV., 2013).

2.2 Integração de Sistemas via Web

Atualmente é muito comum a integração de sistemas de informação de diversos tamanhos e tipos (RICHARDSON, 2007a). Pode-se rapidamente integrar sistemas Web, como por exemplo, o site da Amazon que possui uma integração com sistemas de empresas de cartão de crédito. Pode-se também integrar sistemas Web com aplicativos móveis, como por exemplo, o aplicativo Facebook para dispositivos móveis que permite acessar um perfil via *smartphone* sem ser através de um navegador.

A técnica mais utilizada para a integração de sistemas via Internet é o desenvolvimento de *Webservices*. Segundo o W3C um serviço Web pode ser definido como um sistema de software projetado para suportar a interoperabilidade entre máquinas sobre rede (W3C, 2004). Os *Webservices* são aplicações de serviços na Internet, ou seja, são conjuntos de aplicações auto-descritivas que podem ser publicadas, localizadas e invocadas através da Web (TAMAE, 2005). Uma vez que o *Webservice* seja publicado, outras aplicações podem ter acesso e invocá-lo, conforme ilustrado pela Figura 1.



Figura 1: Funcionamento de um *Webservice*.

Fonte: www.imasters.com.br.

Para requisitar um serviço, o solicitante deve descrever o serviço que deseja e utilizar o provedor de registro para localizá-lo. Após o serviço ser encontrado, a descrição feita pelo solicitante é utilizada para a comunicação entre cliente e servidor. Isto é visível somente para desenvolvedor, ou seja, o usuário depara-se apenas com a interface criada pelo desenvolvedor e não sabe como o processo de busca de informações ocorre (TAMAE, 2005), ou seja, o processo é totalmente transparente para o usuário final.

Um *Webservice* é baseado em tecnologias como HTTP, XML, SOAP, WSDL,

REST, JSON e outros. Sendo que padrões como XML ou JSON, são usados para codificar e decodificar os dados, e protocolos como SOAP e REST podem transportá-los. Eles são a tecnologia ideal para comunicação entre sistemas, integrando sistemas de qualquer porte, sejam eles somente Web ou Web-aplicativos móvel, de maneira fácil, rápida e transparente (W3C, 2002).

Para a Bioinformática, o uso de serviços Web vêm se tornando cada vez mais importante. Os maiores centros de Bioinformática já disponibilizam acesso a seus bancos de dados biológicos através de serviços Web, incluindo o serviço de utilidades do *National Center for Biotechnology Information* (NCBI) (EW BARRETT T, 2009), os serviços do *European Bioinformatics Institute* (EBI) (A VALENTIN F; R, 2007), a API Web do *DNA Data Bank of Japan* (DDBJ) (H, 2003), e a API do KEGG - *Kyoto Encyclopedia of Genes and Genomes* (M GOTO S, 2010). Todos estes serviços são baseados em SOAP ou REST.

Os protocolos de transporte SOAP e REST definem como o serviço Web deverá ser invocado pelo cliente, de acordo com código no servidor. Eles foram criados para encapsular a comunicação entre cliente/servidor e tornam muito mais fácil o desenvolvimento de um serviço pelo programador, já que o mesmo não precisa lidar com detalhes do transporte e da comunicação em baixo nível.

2.2.1 SOAP - *Simple Object Access Protocol*

SOAP é um protocolo de comunicação designado para a Internet, baseado em XML, para codificar informações contidas em mensagens de pedido ou de resposta em serviços Web. As mensagens SOAP são independentes de qualquer sistema operacional, e podem ser transmitidas por diferentes protocolos de Internet, como HTTP ou Multipurpose Internet Mail Extensions (MIME) (SNELL D. TIDWELL, 2001).

Todo o serviço Web baseado em SOAP precisa de um WSDL (*Web Services Description Language*). O WSDL é uma linguagem usada para especificar a localização e as operações ou métodos presentes em um serviço Web (SNELL D. TIDWELL, 2001). Quando o cliente faz uma chamada a um serviço Web é o WSDL que indica onde buscar este serviço. Para tal, normalmente o WSDL utiliza um framework chamado UDDI (*Universal Description, Discovery and Integration*) para auxiliar no uso do serviço Web. O UDDI é usado para descrever e integrar serviços de negócios através da Internet. A descrição das interfaces aos serviços Web é feita através da linguagem WSDL, enquanto o protocolo SOAP é utilizado para a transferência da informação, que por sua vez é dividida em tags com XML.

O funcionamento de um *Webservice* baseado em SOAP é ilustrado pela Figura 2. Após interagir com a interface Web, usada como *front-end*, o usuário envia os dados e a aplicação entra em contato com o provedor UDDI para buscar o método necessário em realizar a conversão. Uma vez associado ao serviço requisitado e sua localização, o provedor retorna para o cliente um arquivo em WSDL, o qual a aplicação completa como uma mensagem. Esta, por sua vez, é enviada ao servidor da aplicação que contém o serviço Web necessário para efetuar a conversão. Com base nas instruções SOAP, o *Webservice* executa sua tarefa, de acordo com os parâmetros recebidos, e envia o resultado obtido com a conversão para o cliente.



Figura 2: Funcionamento de um *Webservice* SOAP.

Fonte: Universidade Atlântica.

2.2.2 REST - *Representational State Transfer*

REST é um estilo de construção de aplicações Web que foi formalmente descrita em (FIELDING, 2000). Um *Webservice* RESTful, não é orientado a conexão, ou seja, não grava o estado da conexão e não utiliza *cookies*. Dessa forma, o aplicativo que utiliza REST, ao invés de SOAP, se torna muito mais leve e com melhor desempenho (RICHARDSON, 2007b). É por este motivo que uma grande parte dos sistemas Web estão migrando de SOAP para REST. O melhor desempenho do REST fica ainda mais evidente quando o cliente é um *smartphone*, o qual possui hardware com menos poder de processamento que um computador pessoal.

A tecnologia foi introduzida primeiramente no ano de 2000, na Universidade da Califórnia, por Roy Fielding (Fielding, 2000). No início, a ideia não mostrou sinais de que seria adotada por muitos. Entretanto, anos após seu surgimento, REST ganharia vários frameworks em que seria trabalhado. Inclusive, devido à sua simplicidade, os serviços

Web RESTful tornaram-se mais utilizados do que aqueles que se utilizam de SOAP e WSDL (RICHARDSON, 2007b).

Em relação a seu funcionamento, o modelo REST especifica restrições que, quando aplicadas em um serviço Web, induzem propriedades vantajosas, como desempenho, escalabilidade e modificabilidade, permitindo que estes serviços tenham um desempenho muito melhor (TYAGI, 2006).

A arquitetura REST considera informação e funcionalidade como recursos que são acessados através de *Uniform Resource Identifiers* (URI), ou seja, links da Internet. Através do modelo, os clientes e servidores trocam informações usando interface e protocolo padronizados. A arquitetura em si é baseada em uma relação Cliente-Servidor, realizando comunicações por um protocolo sem estado (*stateless*), ou seja, cada requisição é uma transação única, sem relação alguma com qualquer outra feita anteriormente.

Serviços Web RESTful usualmente definem os métodos em quatro categorias básicas: GET (obter um recurso), POST (criar um recurso e outras operações), PUT (criar ou atualizar um recurso) e DELETE (deletar um recurso), conforme ilustrado pela Figura 3. Desta forma é muito simples de desenvolver serviços Web baseados em REST.

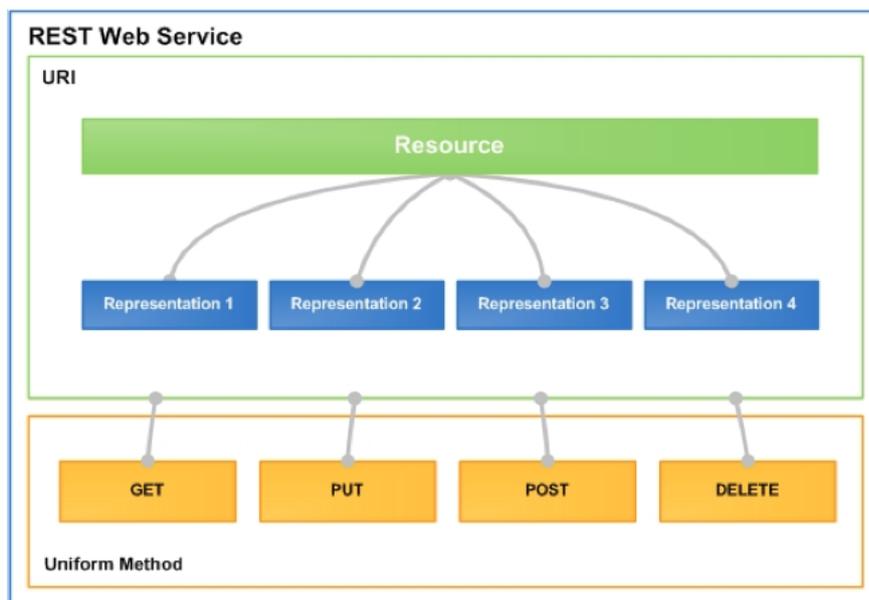


Figura 3: Arquitetura de um *Web Service* RESTful.

Fonte: CrazyLearner.

2.2.3 JSON - *JavaScript Object Notation*

JSON é um formato de texto leve e independente de plataforma, derivado dos literais de Java Script, usado para intercâmbio e serialização de dados. Seu modelo suporta quatro tipos primitivos de variáveis (inteiro, string, booleano e nulo), e dois tipos de variáveis estruturadas (arrays e objetos) (JSON, 1999). Simplicidade é a maior vantagem de JSON, que vêm se mostrando mais eficiente em aplicações Web Ajax do que XML (CROCKFORD, 2008).

Entre outras características que colocam o JSON a frente do XML, está sua maior curva de aprendizado e sua simplicidade quanto a decisões de formatação, uma vez que o formato fornece métodos mais diretos de se mapear aplicações (CROCKFORD, 2008). Ele vêm se tornando uma das técnicas mais utilizadas para troca de dados entre aplicativos móveis e sistemas Web, via *Webservices*, já que encapsula a informação trocada em objetos, serializando e deserializando o mesmo de forma transparente aos programadores.

A estrutura formada por um objeto JSON é similar a um dicionário, utilizando a estrutura chave e valor. A chave é sempre uma string, e precisa estar entre aspas duplas. Já o valor pode ser qualquer um dos tipos primitivos e de variáveis estruturadas já citados. A Figura 4 representa graficamente um sistema Cliente-Servidor no qual é utilizado o protocolo REST com JSON, sendo que a aplicação cliente envia requisições GET, POST, PUT e DELETE com o objeto JSON ao *Webservice* e após a requisição ser processada, o servidor responde com outro objeto JSON à mesma.

2.3 Banco de Dados Biológicos

Uma base de dados biológica é um banco de dados comum com o intuito de organizar a informação biológica e disponibilizá-la de maneira simples aos pesquisadores. Por exemplo, enquanto um banco de dados pode conter um registro sobre um determinado cliente com campos que armazenam nome, idade, local de nascimento e outras propriedades, uma base de dados biológica contém registros associados a sequências de nucleotídeos, descrição do tipo de molécula, o nome científico do organismo e outros (SADEK, 2004).

Com o crescimento rápido da área de Bioinformática, uma enorme quantidade de dados vêm sendo gerada e uma variedade de bases de dados biológicas vêm sendo ofertada por centros de pesquisas renomados na área. Estas bases de dados são em sua grande maioria públicas e cientistas podem não só pesquisar por organismos, como

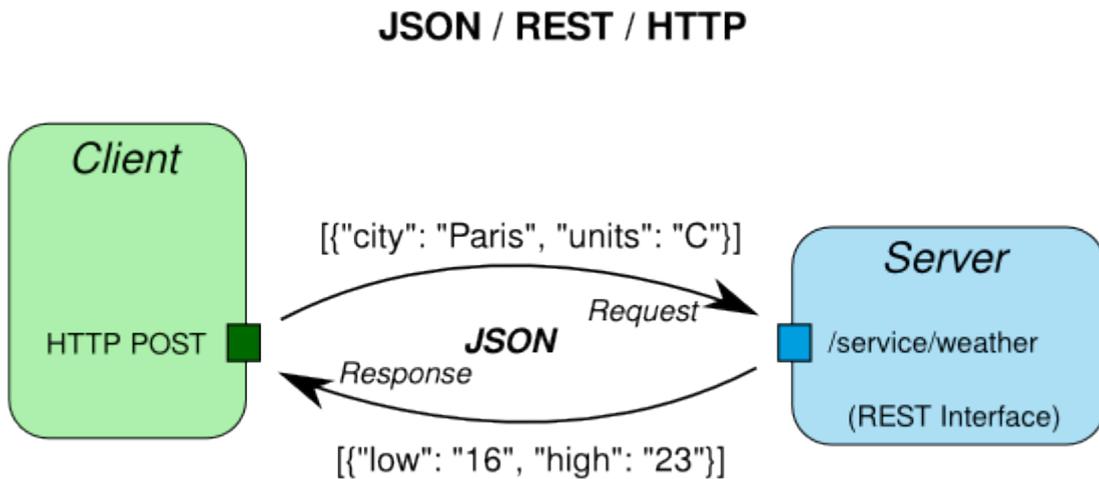


Figura 4: Exemplo REST com JSON.

Fonte: www.safehammad.com/tag/json

também podem cadastrar novos organismos recém descobertos.

A lista de bancos de dados biológicos públicos é grande, entretanto os mais importantes podem ser considerados os seguintes:

- GenBank: é um banco de dados público de sequências de nucleotídeos e anotação de apoio bibliográfico e biológico (BENSON ILENE KARSCH-MIZRACHI, 2005). É mantido pelo *National Center for Biotechnology Information* (NCBI).
- EMBL - ENA: o *European Nucleotide Archive* - ENA (LEINONEN* RUTH AKHTAR, 2010), do *European Molecular Biology Laboratory* - EMBL, é o banco de dados sequências de nucleotídeos europeu.
- DDBJ: o *DNA Data Bank of Japan* (KAMINUMA TAKEHIDE KOSUGE, 2010) é o banco de dados de sequências de nucleotídeos da Ásia. Ele é organizado e mantido pelo *National Institute of Genetics* - NIG.

Estes três bancos de dados biológicos constituem o INSDC *International Nucleotide Sequence Database Collaboration*, cuja a estrutura é ilustrada na Figura 5 e por meio do qual as informações contidas nestas bases são permutadas diariamente.

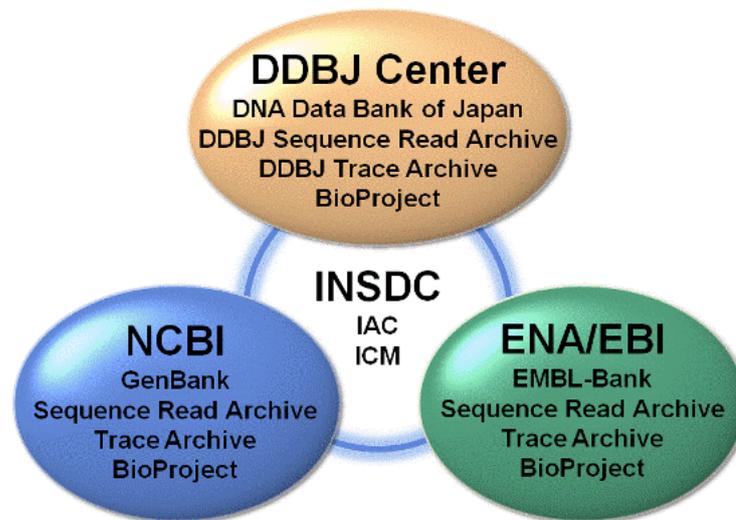


Figura 5: Estrutura do INSDC.

Fonte: (KAMINUMA TAKEHIDE KOSUGE, 2010)

Existem ainda diversas outras bases de dados biológicas, como (M GOTO S, 2010; UNIPROT, 2008; PDB, 2006).

2.4 Ambiente Computacional em Cluster

Cada vez mais vêm aumentando o número de sistemas que necessitam de processamento de dados pesado para um melhor desempenho. Algumas das áreas que demandam processamento de alto desempenho são computação gráfica, meteorologia e Bioinformática.

A primeira solução da comunidade científica foi a construção de supercomputadores, que são sistemas fortemente acoplados (AMERICAN, 2002), porém possuem um custo muito elevado e portanto normalmente não são a primeira escolha dos centros de pesquisa. Uma alternativa mais viável no entanto é a utilização de um ambiente computacional em cluster, também conhecido como sistema fracamente acoplado.

Um ambiente computacional em cluster forma um sistema que une dois ou mais computadores para que os mesmos trabalhem de forma colaborativa com o objetivo de processar diversas tarefas. Estas máquinas dividem entre si as atividades de processamento e executam este trabalho de maneira simultânea.

Uma característica forte de um cluster é que o mesmo deve ser transparente, ou seja, independente da quantidade de máquinas que o compõe ele deve ser visto pelo usuário

ou por outro sistema como um único computador. Desta forma, pode-se considerar que o mesmo soma o processamento e memória RAM individual de cada máquina para executar seus processos de maneira mais rápida, aumentando assim o desempenho (BUYAYA, 2001).

Não há limites na quantidade de máquinas que pode compor um cluster e muitas vezes estas são computadores simples, com processamento mediano. Entretanto, em conjunto, elas formam um sistema de processamento com capacidade suficiente para processar aplicações com maiores necessidades de desempenho e que, se fossem atendidas por supercomputadores, exigiriam investimentos muito maiores. Obviamente, quanto maior a necessidade de processamento de uma certa aplicação, mais poder de processamento as máquinas individuais devem possuir.

Os clusters podem ser classificados, principalmente, em clusters de alto desempenho, clusters de alta disponibilidade e clusters de balanceamento de carga (BUYAYA, 1999).

- Cluster de Alto Desempenho - normalmente atende aplicações que necessitam de um alto poder de processamento. O foco deste tipo de cluster é permitir que o processamento direcionado à aplicação forneça resultados satisfatórios em tempo hábil. Uma área de aplicações que utiliza este tipo de cluster é a de Bioinformática para, por exemplo, realizar o sequenciamento e alinhamento de DNA de organismos.
- Cluster de Alta Disponibilidade - o próprio nome já diz qual o objetivo deste tipo de cluster, o de estar sempre disponível para atender as aplicações. Neste tipo de cluster, não é aceitável que o sistema pare de funcionar. Entretanto, caso aconteça, o tempo de parada deve ser o menor possível, como é o caso de soluções de missão crítica que exigem disponibilidade de, pelo menos, 99,999% do tempo a cada ano. Muitas vezes é aceitável que o sistema apresente alguma perda em seu desempenho, especialmente quando esta situação é devida a esforços para manter a aplicação em atividade.
- Cluster para Balanceamento de Carga - os processos em execução são distribuídos entre as diversas máquinas que compõem o cluster, de forma a tentar um equilíbrio nos esforços dos componentes. O objetivo é que cada máquina componente receba e atenda a uma requisição e não que divida um processo com outras.

O BAK4BIO propõe o uso de um cluster de alto desempenho para que os processos

enviados pelo usuário a partir do dispositivo móvel possam ser executados de uma forma rápida e dentro de um tempo de espera razoável.

2.5 Ruby on Rails

Ruby on Rails é um framework de desenvolvimento Web otimizado para a produtividade sustentável. Permite escrever código fonte de forma elegante, favorecendo a convenção ao invés da configuração (RUBY ON RAILS, 2005). Projeto de código aberto, disponível sob a licença MIT permissiva, e, portanto é gratuito para baixar e usar.

Frequentemente referenciado como Rails ou RoR, foi construído na linguagem de programação Ruby. Seu uso permite o desenvolvimento de aplicações com base no padrão de arquitetura MVC (Model-View-Controller). Segundo (S. THOMAS D., 2011), tornou-se a escolha natural para o desenvolvimento de uma ampla gama de aplicativos para a Web 2.0.

Com uma arquitetura elegante e compacta, explora a maleabilidade da linguagem Ruby, ao efetivamente criar uma linguagem de domínio específico para escrever aplicações Web. Rails também se adapta rapidamente as novas demandas das tecnologias. Por exemplo, o Rails foi um dos primeiros a consumir e implementar o estilo de arquitetura REST para a organização de aplicações Web.

Finalmente, o framework dispõe de uma comunidade vasta e diversificada. Como resultado, tem-se centenas de colaboradores, diversas conferências, um número enorme de *plugins* e bibliotecas com soluções para problemas específicos. Além disso, conta com uma rica variedade de blogs informativos e uma vasta lista de discussões em fóruns sobre a tecnologia.

2.6 Controle de versão - Git

No desenvolvimento de qualquer projeto de software, muitas vezes se faz necessário o controle de versões e *backup* do sistema. No projeto BAK4BIO esta foi uma preocupação constante, manter um versionamento e cópias do sistema de maneira segura e eficaz.

O Git é um sistema gratuito e livre, distribuído sob os termos da versão 2 da GNU General Public License (GPLv2), para controle de versão em projetos de software.

Escrito na linguagem de programação C, consegue reduzir a sobrecarga no tempo de execução comparados com as linguagens de alto nível, como Java. Por não depender de acesso a uma rede ou a um servidor central para seu correto funcionamento, quase todas as suas operações são realizadas localmente, dando-lhe uma vantagem enorme em relação aos sistemas centralizados que constantemente tem que se comunicar com um servidor. Devido ao seu desempenho e facilidade de uso optou-se por adotar o Git como o sistema de versionamento do projeto BAK4BIO.

2.7 Computação em nuvem

Segundo (A.T VELTE T.J, 2009) a computação em nuvem é uma construção que permite acessar aplicativos os quais realmente residem em um local diferente do seu computador ou outro dispositivo conectado à Internet, na maioria das vezes, em um datacenter. O autor acrescenta (texto traduzido):

”A beleza da computação em nuvem é que uma outra empresa hospeda sua aplicação. Isto significa que eles são responsáveis pelo custo dos servidores, atualização de software e toda a infra-estrutura relacionada.”

Desta forma fica claro que a preocupação do desenvolvedor é apenas a de realizar seu projeto, toda a infra-estrutura de apoio e produção fica a cargo de uma empresa terceira a ser contratada para o serviço. Portanto, a partir de um computador ou dispositivo móvel com acesso a Internet, é possível acessar informações, arquivos e programas em um único sistema e independente de plataforma, conforme ilustra a Figura 6.

Várias grandes empresas já vem explorando esta tecnologia, alguns exemplos são: a Apple, com seus *streamings* de áudio e vídeo via iTunes; o Google com a possibilidade de editar documentos online e armazená-los no Google Docs; a Microsoft com o Windows Azure; e a Amazon com seu datacenter. Um outro caso de sucesso é o Dropbox, um dos mais usados para armazenamento e compartilhamento de arquivos e que serviu de inspiração para a construção do módulo BAK4BIO Box, um espaço de armazenamento de arquivos com sequências de DNA e nucleotídeos, contendo dados de entrada e saída de processos ligados a Bioinformática.

De acordo com (SHARGI, 2009) os serviços mais ofertados por empresas que mantém uma infra-estrutura de computação em nuvens são SaaS (*Software as a Service*), o qual um provedor fornece aplicações ao usuário sem que o mesmo precise instalar sistemas

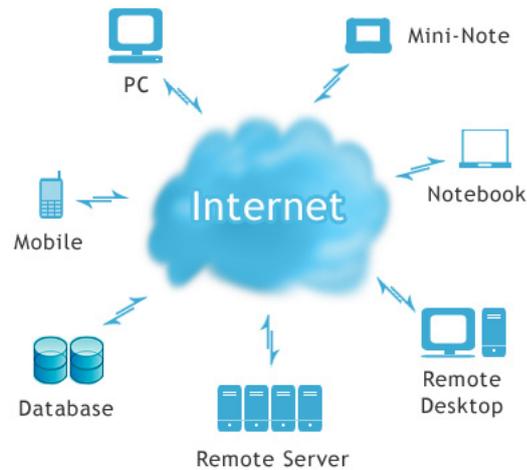


Figura 6: Exemplo da arquitetura presente em computação nas nuvens.

Fonte: (TARGET TRUST, 2009)

em seu computador; PaaS (*Platform as a Service*), o qual permite que desenvolvedores utilizem plataforma de desenvolvimento bastante poderosas de forma remota, como o próprio Windows Azure; e IaaS (*Infrastructure as a Service*), utiliza-se servidores para hospedagem de sistemas e aplicações.

As principais características de um ambiente de computação em nuvens são provisionamento dinâmico de recursos sob demanda, com mínimo de esforço; escalabilidade; uso de *utility computing*, onde a cobrança é baseada no uso do recurso ao invés de uma taxa fixa; visão única do sistema; e distribuição geográfica dos recursos de forma transparente ao usuário.

2.8 Padrões de Projeto

Segundo (GAMMA R. HELM, 1998), padrão de projetos (em inglês *design patterns*) é uma solução repetitiva para uma questão recorrente na construção de um software. É uma descrição ou modelo de como resolver um problema que pode ser utilizada em situações diferentes. O autor acrescenta que um padrão possui quatro elementos essenciais: o nome, que pode ser usado para descrever um padrão de projeto; um problema ou cenário, que descreve quando aplicar o padrão; a solução, que informa os elementos que compõem o padrão, seus relacionamentos, responsabilidades e colaborações; e as consequências, que são resultados e vantagens e desvantagens de se aplicar um determinado

padrão.

2.8.1 MVC

Conforme (GAMMA R. HELM, 1998) o MVC é um dos padrões de projeto, composto por três camadas ou objetos (*Model*, *View* e *Controller*), o qual permite dividir as funcionalidades do sistema em diferentes responsabilidades para aumentar a flexibilidade e reutilização dos códigos fontes. Conforme apresentado na Figura 7.

a) *Model*

Responsável por representar as informações presentes no software;

b) *View*

Responsável por apresentar as informação aos usuários finais;

c) *Controller*

Responsável por controlar o fluxo e os valores das informações do sistema conforme os seus respectivos comportamentos.

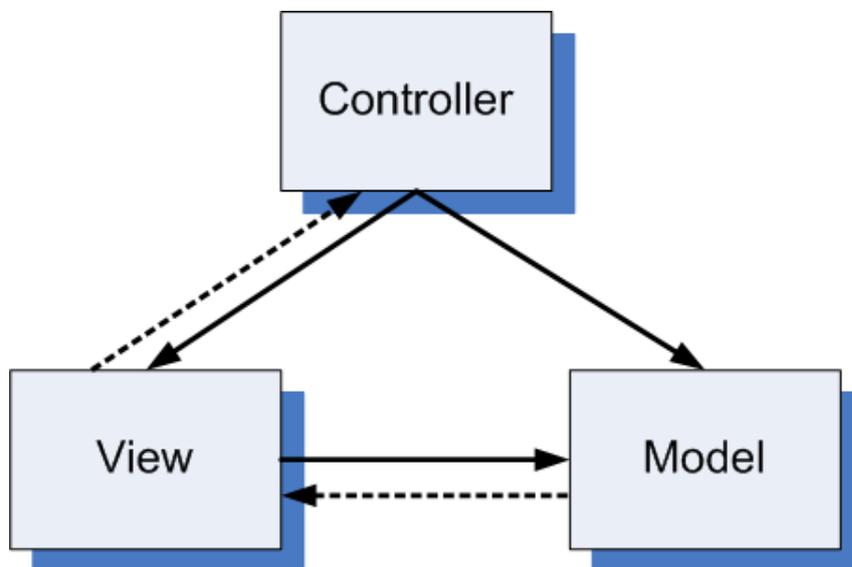


Figura 7: Padrão de projetos MVC

Fonte: (HARVARD, 2010)

2.9 Trabalhos Relacionados

Em estudos preliminares realizados, foram identificados alguns trabalhos relacionados ao tema e às tecnologias neste projeto utilizadas. Esta seção visa apresentar os mais importantes trabalhos correlatos para auxiliar na construção de uma visão do estado da arte. Ela está dividida em três tópicos distintos: Serviços Web, que apresenta outros trabalhos que utilizam serviços Web para integração de sistemas de informação para Bioinformática; Dispositivos Móveis, que apresenta outros trabalhos que desenvolveram aplicativos ou soluções para *smartphones*; e Supervisão de Workflow, que apresenta trabalhos que desenvolveram sistemas supervisórios para gerenciamento de processos executados em ambientes de servidores remotos ou clusters de processamento.

2.9.1 Serviços Web

Esta seção apresenta alguns dos trabalhos relevantes que utilizam a tecnologia de serviços Web para prover acesso, na maioria dos casos, aos serviços ofertados pelos centros de pesquisa de Bioinformática, sejam estas bases de dados ou ferramentas computacionais.

2.9.1.1 TogoWS

O projeto TogoWS ¹, apresentado em (KATAYAMA; TAKAGI, 2010), propõe uma plataforma de serviços Web capaz de integrar de maneira transparente o acesso às principais bases de dados biológicas, desta forma melhorando consideravelmente a usabilidade e facilitando o desenvolvimento de sistemas que necessitem utilizar estas bases.

Durante o desenvolvimento do TogoWS percebeu-se que os serviços Web mais usuais eram projetados para procurar e recuperar as entradas de banco de dados mantidos em cada centro de pesquisa. Portanto, projetou-se um serviço Web baseado em REST para acessar os recursos das bases de uma forma unificada, com a notação URI intuitiva para pesquisar, recuperar, analisar e converter as entradas de cada um dos bancos de dados. Além disso, foi também desenvolvido um serviço de Web único, baseado em SOAP, que filtra o acesso a serviços Web fornecidos pelos centros de pesquisa (DDBJ e PDBj) japoneses para resolver vários problemas de incompatibilidade encontrados nestes serviços. A Figura 8 ilustra a arquitetura do projeto TogoWS, note que a partir de serviços Web baseados em REST e SOAP, um sistema cliente pode acessar diversas bases de dados biológicas.

¹Togo é uma palavra japonesa que significa integração

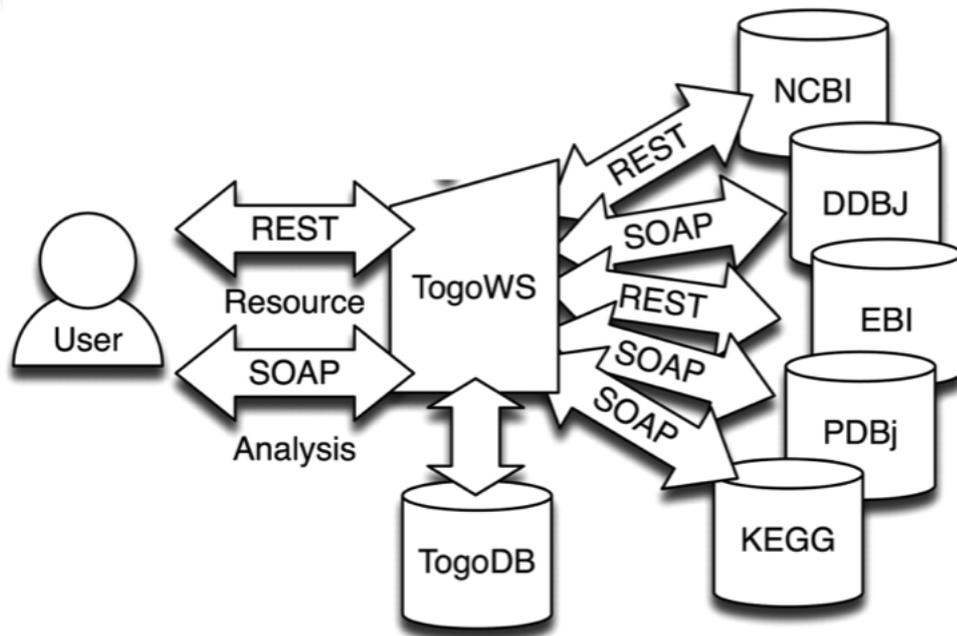


Figura 8: Arquitetura do projeto TogoWS.

Adaptado de (KATAYAMA; TAKAGI, 2010)

Como o TogoWS atua de certa forma como um *proxy* de acesso a diversos serviços de centros de pesquisa biológicos, facilitando o acesso e filtrando incompatibilidades, ele vêm sendo utilizado por diversos outros sistemas clientes desenvolvidos. O BAK4BIO, aproveitando-se também deste serviço já fornecido, utiliza o TogoWS para que seu cliente no dispositivo móvel possa ter acesso transparente às bases de dados biológicas do NCBI, EBI, DDBJ, PDBj e KEGG.

2.9.1.2 A Multi-Protocol Bioinformatics Web Services e HitKeeper

Um outro sistema que provê acesso a múltiplos serviços Web baseados em SOAP e REST é o projeto apresentado em (PAGNI; STOCKINGER, 2008). Para fornecer acesso via múltiplos padrões e protocolos de *Web Services*, os autores desenvolveram em Perl um sistema de acesso a serviços Web que utilize a ferramenta de Bioinformática *HitKeeper* (HAU; PAGNI, 2007).

O *HitKeeper* é também um exemplo de ferramentas de Bioinformática que utilizam serviços Web. Ele suporta e implementa o *WS-I Basic Profile 1.1*, possui também serviços Web baseados em REST e formulários de acesso baseados em HTML.

O diferencial do projeto (PAGNI; STOCKINGER, 2008) é que constrói um sis-

tema Web sobre o HitKeeper que pode ser acessado por ambientes de computação em Grid e Cluster. Os autores não apresentam uma análise de escalabilidade do serviço, no entanto este estudo também serviu como base para a proposta de desenvolvimento do módulo BAK4BIO Cluster, que recebe requisições remotas para a execução de processos.

2.9.1.3 WS-BioZard

Um outro exemplo de como os serviços Web estão influenciando e sendo utilizados na Bioinformática é o trabalho apresentado em (WANG; KISSINGER, 2008), o WS-BioZard. Este é um framework semi-automático para descoberta de serviços Web, composição de serviços e mediação de dados. Ele utiliza anotações semânticas para WSDL e ontologias para adicionar semântica aos serviços Web.

O ponto chave de contribuição do WS-BioZard é fornecer uma ferramenta de descoberta de serviços com uma interface de usuário sofisticada e uma interface semi-automatizada para a composição de novos serviços Web, funcionando como se fosse um compilador de serviços Web.

Desta forma, o WS-BioZard é uma plataforma que auxilia não-programadores, segundo os autores focada em biólogos, a desenvolver serviços Web através de uma interface gráfica amigável. Além disto, após o desenvolvimento do serviço Web, a plataforma busca erros no processo de desenvolvimento, e se não encontrar disponibiliza o serviço para o público.

2.9.2 Dispositivos Móveis

Apesar de processos ligados a Bioinformática normalmente precisarem de processamento pesado, muitas vezes em grids ou clusters computacionais, esta seção visa mostrar que já há alguns trabalhos que exploram o potencial de dispositivos móveis para prover soluções com mobilidade para a Bioinformática, sejam estes aplicativos locais, acessando dados no próprio dispositivo, ou aplicativos que se integram à outros sistemas via Internet.

2.9.2.1 BioWAP

O BioWAP, apresentado em (RIIKONEN J. BOBERG; VIHINEN, 2001), é, segundo os autores o primeiro trabalho a trazer a ideia de mobilidade para a Bioinformática. O projeto possibilita acesso a diversas bases de dados e ferramentas de análise de dados

biológicos a partir de um telefone celular. O aplicativo desenvolvido fornece acesso a um total de 20 ferramentas de análise e bases de dados.

Apesar de extremamente inovador para sua época, ano de 2001, este serviço atualmente caiu em desuso devido a tecnologia ultrapassada da época. Para o desenvolvimento do BioWAP, foi utilizado o que havia de melhor em termos de padrão de comunicação de dados no início dos anos 2000, ou seja, a tecnologia WAP (*Wireless Application Protocol*). WAP foi desenvolvido para prover serviços equivalentes a um navegador Web com alguns recursos específicos para serviços móveis, não permitindo o uso de páginas Web baseadas em HTML. Devido a este fato, durante sua existência, sofreu com a pouca atenção dada pela mídia e foi muito criticado por suas limitações.

Como WAP era bastante limitado em termos do que se podia programar, o BioWAP apresentava limitações em relação a sua interface gráfica e usabilidade. Além disto, também devido a tecnologia da época, sofria limitações em relação a velocidade de acesso à informações necessárias.

Independente de diversas limitações, vale ressaltar que o BioWAP é considerado uma grande inovação no seu tempo, sendo uma das primeira soluções a proporcionar mobilidade ao acesso a informações biológicas, e portanto deve ser lembrado.

2.9.2.2 *SimGene* e *SimAlign*

Desenvolvidos e mantido pelo Japan Bioinformatics KK, o *SimGene* (SIMGENE, 2011) e o *SimAlign* (SIMALIGN, 2012) são aplicativos móveis, multiplataforma (iOS e Android), projetados para biólogos, bioinformatas e pesquisadores médicos.

O *SimGene* oferece acesso transparente, via serviços Web, às bases de dados Simbiot, Ensembl, NCBI, Gene Ontology, Pathways KEGG e PubMed. O aplicativo busca informações de anotação de mais de 30 espécies com base no símbolo do gene fornecido inicialmente pelo usuário. A partir de então oferece um navegador integrado com informações sobre genes, transcrições, *exons* e SNPs, além de informações de referência cruzada para as bases do NCBI, Ensembl, RefSeq e UniProt.

O *SimAlign* se comunica com o *BLASTNCBI+* para alinhar uma sequência fornecida para uma variedade genômica, RNA e bancos de dados de proteínas. O aplicativo salva todas as consultas e permite aos usuários facilmente recordar e reenviar pesquisas anteriores.

2.9.2.3 Aplicativo FlyExpress

Um outro exemplo de como os aplicativos móveis estão influenciando as áreas de Bioinformática e Biotecnologia é o aplicativo *FlyExpress*, apresentado como uma *Application Note* em (KUMAR K. BOCCIA; YE, 2012). Os autores do projeto exploram as tecnologias móveis para distribuir uma coleção crescente, atualmente mais de 100.000, de imagens padronizadas de hibridização *in situ* contendo padrões espaciais de expressões genéticas de *Drosophila Melanogaster* (mosca da fruta).

O aplicativo exibe o padrão de expressão do gene selecionado para diferentes projeções visuais e exibe-os de acordo com suas fases de desenvolvimento, o que mostra a progressão da expressão espacial de um gene ao longo de seu desenvolvimento.

2.9.2.4 Hematopoietic Expression Viewer

O *Hematopoietic Expression Viewer*, apresentado como uma *Application Note* em (JAMES M. M. RAO, 2012), é mais um projeto que aposta na vantagem da mobilidade para acessar ferramentas ou informações biológicas. Os autores desenvolveram um protótipo de um aplicativo móvel para a exibição de dados de expressão genética usando a plataforma iOS. O aplicativo tenta equilibrar o acesso as informações entre uma maneira local, usando estruturas de armazenamento de dados nativas do iOS, e acesso via Internet, quando há a necessidade de consultar dados não armazenados localmente.

2.9.3 Supervisão de *Workflows*

Um problema na utilização de grids e clusters computacionais é a organização da execução de algum processo no ambiente computacional, visto que este pode conter diversas interações entre programas e banco de dados armazenados em diferentes lugares. Para auxiliar no gerenciamento destes processos e visualização do estados dos mesmos (que fase de execução se encontra) pode-se utilizar os supervisores de *workflow*. Esta seção apresenta alguns dos principais sistemas de supervisão de *workflows*.

2.9.3.1 SciCumulus

O SciCumulus, proposto em (OLIVEIRA E. OGASAWARA, 2012) é um sistema supervisor projetado para distribuir e controlar a execução paralela de workflows em um ambiente de nuvem. O sistema cria um ambiente de cluster virtual para a execução de

processos em um conjunto distribuído de máquinas virtuais (VMs).

O processo se inicia com a reserva de recursos para execução em paralelo na nuvem das tarefas necessárias. Logo após, o SciCumulus cria e gerencia a execução paralela de atividades em uma ou mais VMs instanciadas na nuvem. Durante a execução dos processos, dados de entrada e dados de proveniência são consumidos e gerados pela execução paralela das atividades. Desta forma, o SciCumulus provê um ambiente computacional propício para o paralelismo em *workflows* com coleta de proveniência distribuída, possibilitando consultas aos dados do experimento em tempo real.

A desvantagem é que o sistema não fornece funcionalidades de monitoramento e notificação que auxiliem o usuário a acompanhar o estado da execução de seus processos sem que necessite estar fisicamente em algum terminal, como é o caso do BAK4BIO que possibilita o acompanhamento das tarefas a partir de um dispositivo móvel.

2.9.3.2 SciLightning

O objetivo do trabalho (PINTAS D. OLIVEIRA, 2012) é apresentar um sistema de monitoramento de execução de *workflows* que pode ser acoplado às soluções para execução de processos já existentes. O sistema integra uma solução supervisória com clientes para dispositivos móveis e redes sociais. Os autores demonstram um caso de uso acoplando o sistema supervisor SciLightning ao gerenciador de *workflows* SciCumulus, desta forma incrementando o SciCumulus com a funcionalidade de supervisão em tempo real.

2.9.3.3 Submissão e Monitoração de Tarefas em Grid

O trabalho apresentado em (BORGES; DANTAS, 2006), visa, assim como o sistema supervisor do BAK4BIO, auxiliar no gerenciamento de atividades envolvidas na execução de um processo. Para a realização da supervisão foi empregado o conceito de *workflow* em um ambiente de grid computacional.

O sistema supervisor emprega uma ontologia descrevendo os recursos utilizados em cada atividade que compõem um determinado *workflow* e qual problema específico o mesmo pode resolver. Desta forma, fornecendo conhecimento ao usuário para que ele escolha um *workflow* que possa resolver seu problema de uma maneira mais adequada, semelhante às ideias apresentadas em (YU X. BAI, 2004).

3 MATERIAIS E MÉTODOS

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

3.1 Visão Geral

O presente trabalho descreve a construção do framework para desenvolvimento de soluções digitais aplicados a área de Bioinformática e afins.

O objetivo é prover uma infra-estrutura robusta e enxuta, capaz de interligar diferentes soluções em uma única plataforma de desenvolvimento, bem como explorar a carência de ferramentas para dispositivos móveis na área. A solução, chamada de BAK4BIO: Brazilian Army Knife for Bioinformatics, é composta por cinco módulos:

- a) BAK4BIO Server
Autorização e autenticação; comunicação entre módulos.
- b) BAK4BIO Box
Armazenamento de arquivos na nuvem; entradas e saídas de processos.
- c) BAK4BIO Cluster
Execução de rotinas complexas; paralelismo de operações.
- d) BAK4BIO Droid
Canal de comunicação; via dispositivos móveis.
- e) BAK4BIO Web
Canal de comunicação; via Web.

O fluxo de requisições se inicia a partir do módulo Droid (*smartphone*) ou do Web. O usuário solicita acesso a alguma informação remota, podendo esta ser um arquivo no

Box, ou uma informação das bases de dados biológicas, ou o início de um processo a ser executado pelo Cluster. Esta requisição é enviada ao módulo Server, que identifica o tipo de requisição e a envia para o destino correto, conforme ilustrado pela Figura 9.

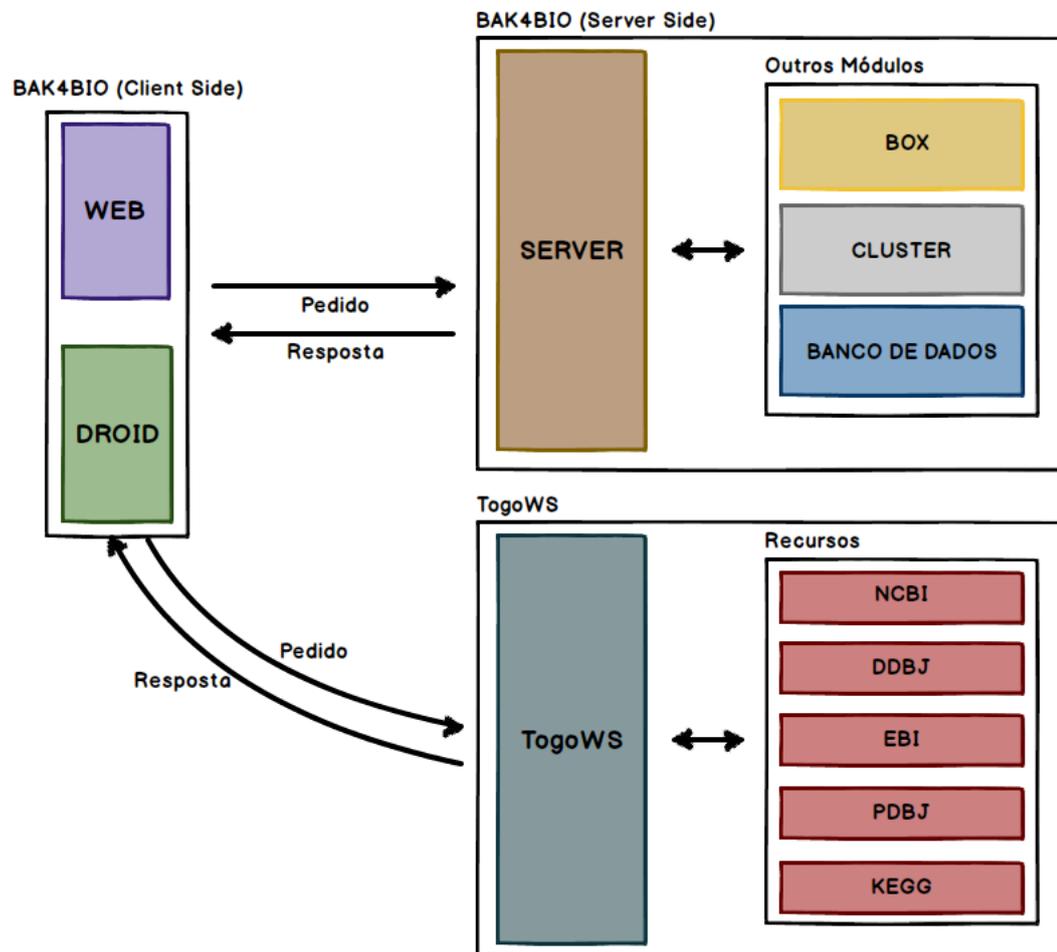


Figura 9: Fluxo de requisições no BAK4BIO.

Fonte: o autor (2013).

Conforme demonstrado pela Figura 10, a arquitetura visa o desacoplamento entre os módulos e divisão de responsabilidades. Isso significa dizer que o framework BAK4BIO facilita a manutenção e o incremento de novas funcionalidades, bem como o reaproveitamento de código fonte a outros projetos da comunidade de desenvolvedores em Bioinformática.

3.2 BAK4BIO Server

Nessa seção é apresentado o módulo responsável pela autenticação, autorização e comunicação entre os módulos do BAK4BIO.

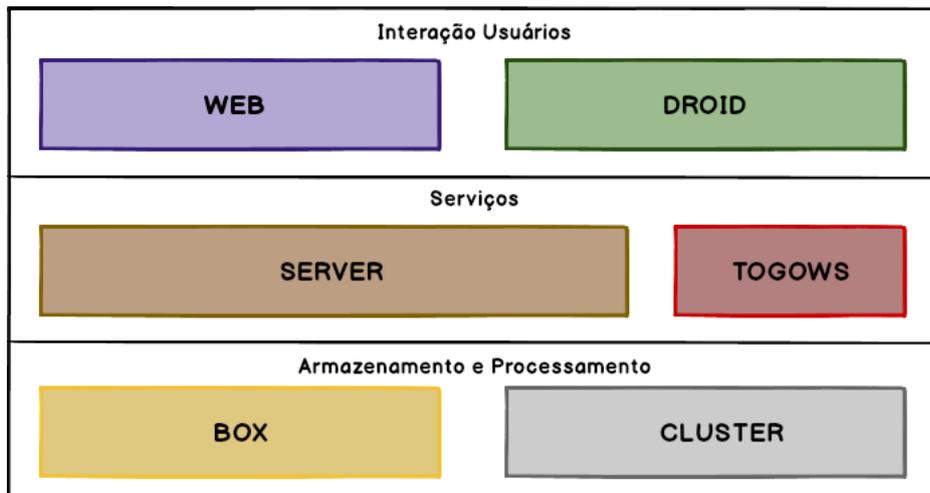


Figura 10: Arquitetura do Framework BAK4BIO.

Fonte: o autor (2013).

3.2.1 Autenticação e autorização

Autenticação e autorização são processos fundamentais para garantir a segurança dos sistemas Web. A primeira se refere a validar o usuário, geralmente utilizando um login e senha. Entretanto, há também outros métodos, como uso de cartão inteligente (*smart card*), biometria e certificados digitais. A segunda, tem como objetivo descobrir se uma pessoa, uma vez autenticada, tem permissão para acessar quais áreas do sistema e executar quais operações.

Neste trabalho foi utilizado a biblioteca Devise (PLATAFORMATEC, 2013), idealizada e desenvolvida por um time de brasileiros. Segundo o portal (RUBY TOOLBOX, 2013) é a ferramenta de autorização com mais adeptos dentre os desenvolvedores *ruby* no mundo. Fornece uma arquitetura flexível e de fácil customização. A lista abaixo apresenta algumas funcionalidades presentes na solução:

- Criptografia das senhas na base de dados: Realiza o armazenamento de senhas de forma segura com o uso do método Bcrypt, do tipo hash e tem como base o Blowfish.
- Token de autenticação: Uma vez autenticado, substitui as informações de login e senha nas próximas requisições. Importante para garantir a segurança na comunicação entre os dispositivos móveis e o servidor.
- Confirmação de conta: Garante a legitimidade do usuário cadastrado por meio da sua conta de email.

- d) Recuperação de senha: Permite a recuperação da senha cadastrada através de recebimento de email automático gerado pelo BAK4BIO.
- e) Cadastro de novo usuário: Possibilita a ingressão de novas contas para acesso as funcionalidades restritas, ou seja, os processos que precisam de autenticação.
- f) Gerenciamento de sessão: Realiza o controle do tempo de inatividade dos usuários. Ao exceder o tempo devido, requer nova autenticação.
- g) Validações de email e senha: Evita realizar cadastros com emails inválidos e ajuda os novos usuários a realizarem senhas seguras para a sua conta no BAK4BIO.
- h) Bloqueio de conta conforme a quantidade de tentativas inválidas de login: Limita a tentativa de acesso a conta. Inibe o usuário mal intencionado de realizar por força bruta o descobrimento das senhas de usuários cadastrados.

A Figura 11 ilustra de forma geral, o processo de autenticação envolvido na solução BAK4BIO. Ao entrar na área de operações restritas, o usuário é convidado a realizar a sua autenticação através das suas informações pessoais de email e senha (passo 1). Ao receber o pedido, o servidor encaminha-o ao Devise (passo 2), que por sua vez valida as informações contra a base de dados (passo 3). Ao finalizar o procedimento com sucesso, é retornado (passo 4) a confirmação da conta e realizado o armazenamento da autenticação na sessão do navegador (passo 5) ou na entidade de usuário (passo 5).

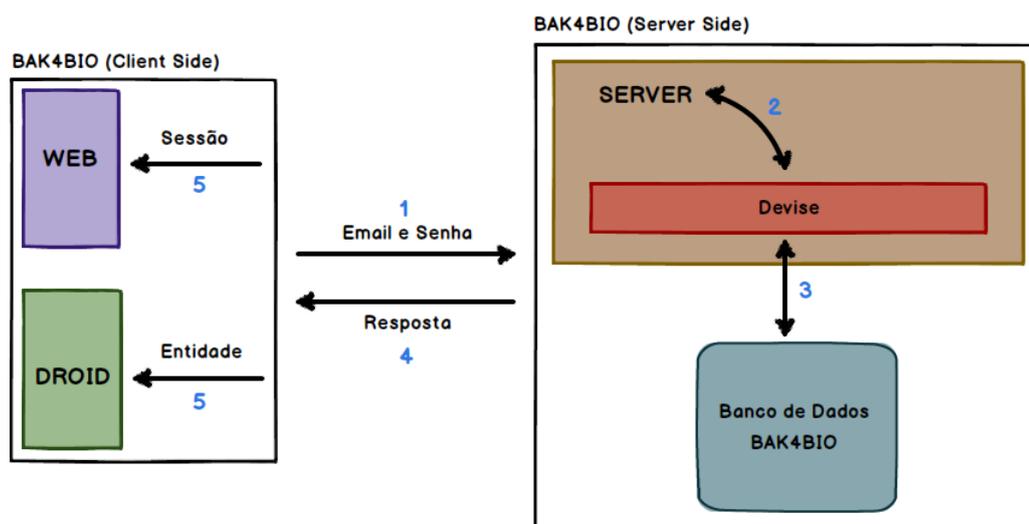


Figura 11: Visão geral do processo de autenticação.

Fonte: o autor (2013).

Após realizada a autenticação, o usuário está apto a executar as operações. Entretanto, isso não significa necessariamente que o usuário em questão tenha habilidades para executar ou visualizar qualquer informação. É nesse momento que entra a biblioteca Cancan, processo para garantir que o usuário (então autenticado) tenha permissão para executar tais rotinas.

O Cancan (CANCAN, 2009) é uma biblioteca de autorização para Ruby on Rails para categorizar os usuário conforme o seu perfil e conseqüentemente a porcentagem de acesso aos sistemas. A Figura 12 demonstra de forma geral, o processo envolvido no processo de autorização do envolvido no BAK4BIO. Uma vez autenticado no sistema, toda requisição feita ao módulo Server (passo 1) é verificado junto ao cancan (passo 2) se para realizar a operação em questão o usuário possui acesso (passo 3). Ao confirmar a autorização, é feito o processamento e retorno das informações (passo 4). Caso contrário, ou seja, quando o usuário não tenha acesso, é redirecionado a uma página informando a restrição de acesso (passo 4).

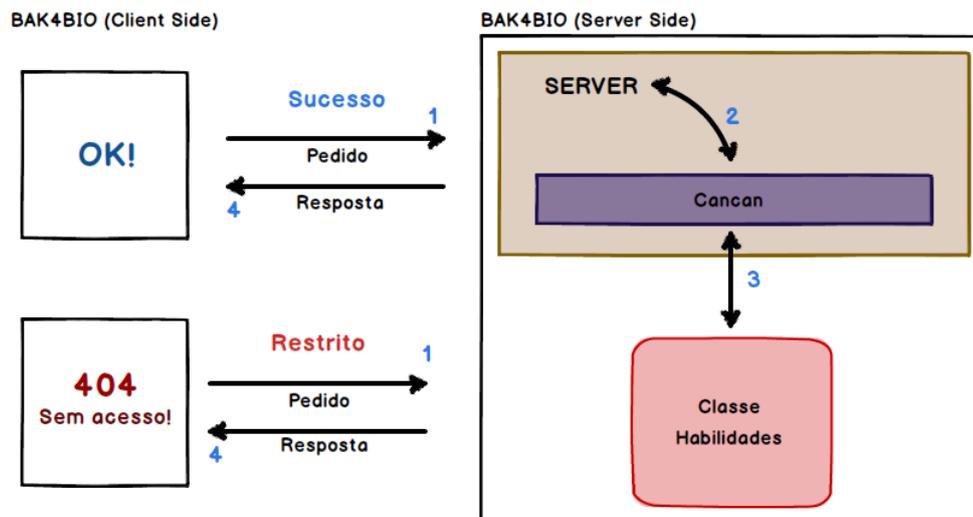


Figura 12: Visão geral do processo de autorização.

Fonte: o autor (2013).

3.2.2 Proxy

Responsável por centralizar o tratamento das requisições vindas dos módulos Web e Droid (Figura 13), assume a função semelhante a um proxy por realizar o processo de cache de informações, bem como determinar o controle sobre a navegação do sistema. As suas principais funcionalidades são apresentadas abaixo:

- a) Centralização da segurança (autenticação e autorização): Promove a segurança das informações dos usuários cadastrados.
- b) Cache de informações: Realiza a otimização de resultados ao identificar um pedido de requisição igual e já processado anteriormente.
- c) Controle sobre a navegação das páginas Web: Determina a navegação conforme o tipo de processamento, bem como a mensagens e tratamentos de erros provenientes de informações inválidas.
- d) Acesso a base de dados: Responsável pelo acesso e gerenciamento das informações presentes na base de dados do BAK4BIO.
- e) Segurança sobre a comunicação entre os módulos: Inibe usuários mal intencionados de simular pedidos a módulos isolados da solução, ou seja, evita do mesmo se passar por algum módulo da solução.

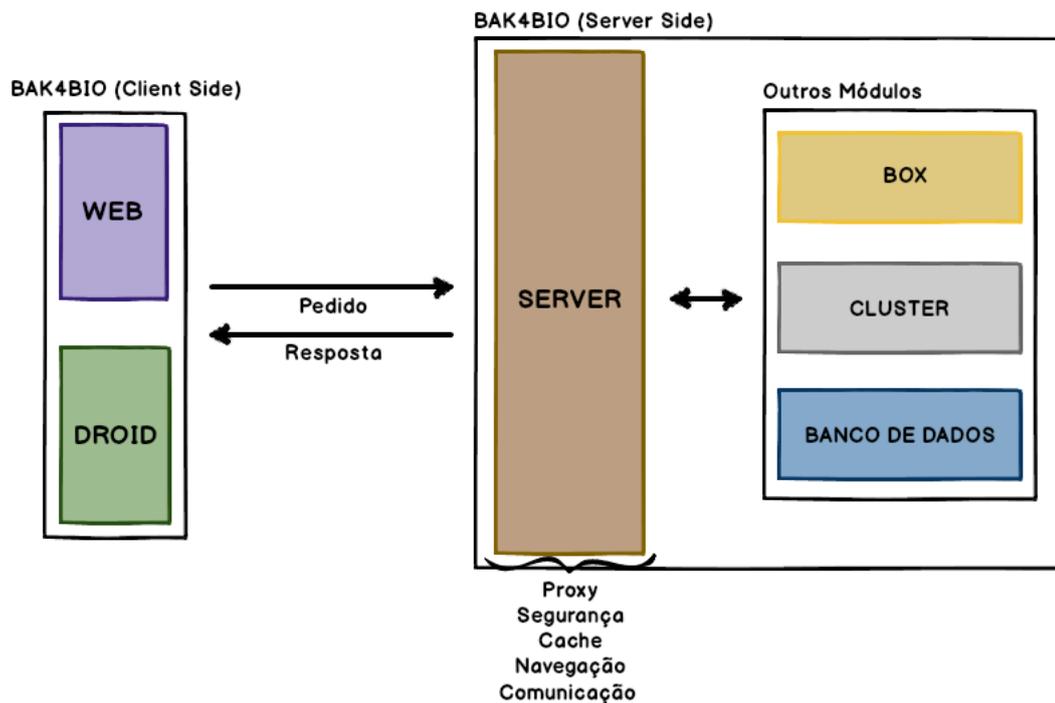


Figura 13: Comunicação entre os módulos.

Fonte: o autor (2013).

3.3 BAK4BIO Box

Nessa seção é apresentado o módulo Box, responsável pelo armazenamento de arquivos provenientes de entradas e saídas dos processos envolvidos no framework BAK4BIO.

Construído com base nos conceitos de computação em nuvem, especialmente no que se diz respeito a IaaS, tem como objetivo fornecer acesso as informações e arquivos armazenados pelo usuário, em qualquer lugar, a qualquer momento, desde que haja acesso à Internet, de forma centralizada e sem a necessidade de instalação de aplicações auxiliares. A lista abaixo apresenta as principais funcionalidades do módulo:

- a) Executar operações de upload e download de conteúdos;
- b) Armazenar as informações referentes aos conteúdos;
- c) Servir como um módulo de armazenamento, mantendo os arquivos de dados em disco;
- d) Assegurar que apenas o proprietário da informação tenha acesso a elas.

O BAK4BIO Box funciona em conjunto com os módulos Web e Server. Enquanto, o primeiro fornece a interface necessária para o gerenciamento das informações aos usuários via navegador, o segundo se encarrega da segurança, bem como a realização da transferência do conteúdo ao módulo Box, conforme ilustrado na Figura 14.

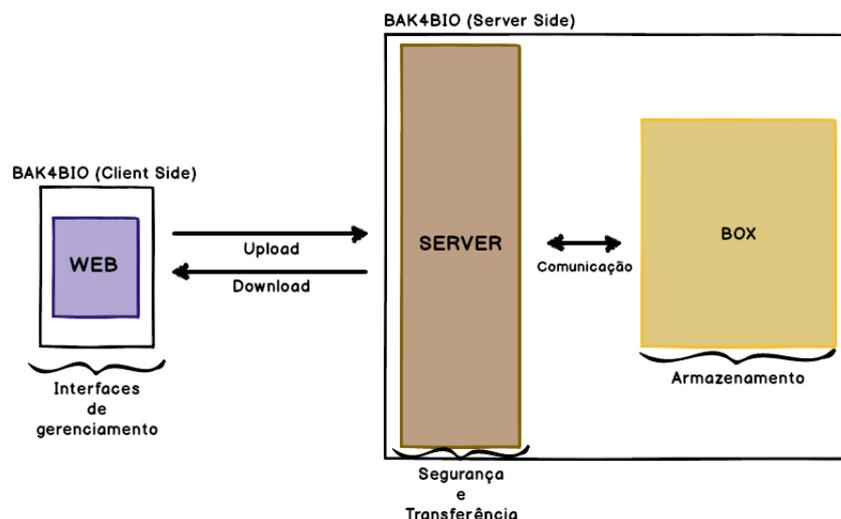


Figura 14: Procedimento de armazenamento do módulo Box.

Fonte: o autor (2013).

A interação com o usuário ocorre através do módulo Web. O processo se inicia ao realizar a devida autenticação e autorização no framework, conforme ilustrado na Figura 15. Após isso, é possível selecionar a opção Box e é apresentado os eventuais arquivos que já tenham sido cadastrados anteriormente (Figura 16). A partir disso, é possível realizar a devida manutenção dos conteúdos presentes no módulo Box.

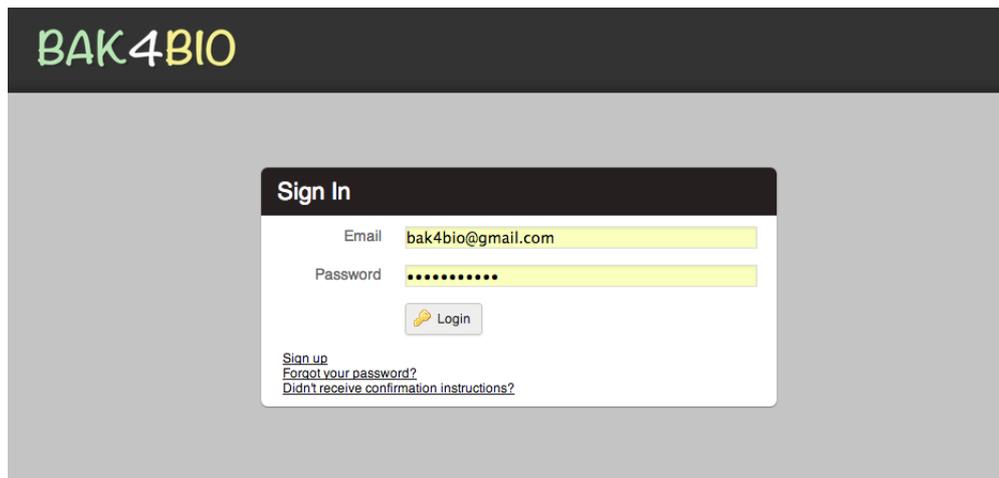


Figura 15: Interface de acesso ao módulo Web.

Fonte: o autor (2013).

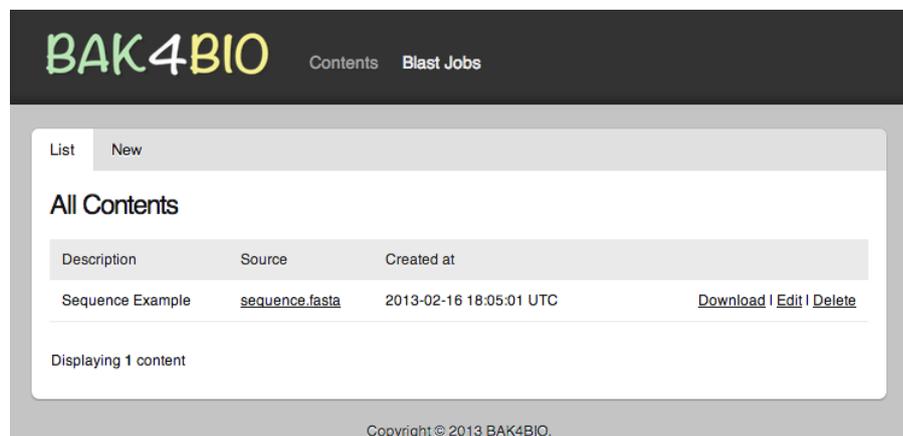


Figura 16: Interface de listagem aos arquivos.

Fonte: o autor (2013).

Com os arquivos presentes na solução, as informações já ficam acessíveis em qualquer local com acesso à Internet e aptas a serem utilizadas em outros processos existentes no framework, como por exemplo a operação de BLAST, na qual é exigida um arquivo de entrada com as informações da sequência genética do organismo a ser processado.

3.4 BAK4BIO Cluster

Nessa seção é apresentado o módulo Cluster, responsável pela execução e paralelismo de operações complexas do BAK4BIO.

Composto por uma arquitetura simplificada e escalável, torna possível a integração de vários projetos com demandas de alto poder computacional. Assume o papel de consumidor de tarefas pendentes existentes no BAK4BIO Server. Tarefas que são produzidas pelos usuário finais (através dos módulos Web e Droid) ou pelo próprio framework, denominadas de tarefas internas. A Figura 17 exemplifica o fluxo geral das atividades e a comunicação dos módulos envolvidos no processo. Note que o pedido do usuário é recebido pelo servidor, armazenado e executado no cluster. A toda mudança de status do pedido, notificações são enviadas ao cliente.

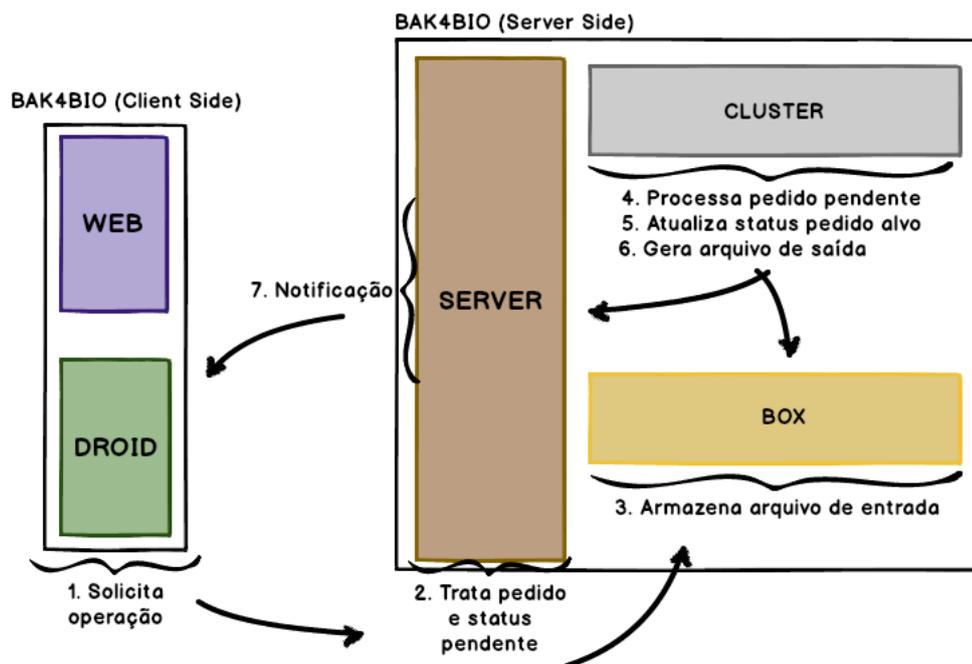


Figura 17: Fluxo de atividade do BAK4BIO Cluster.

Fonte: o autor (2013).

Para consumir as tarefas pendentes, foi desenvolvido um software, usando a tecnologia Java, responsável em verificar de tempos em tempos os novos pedidos disponíveis no módulo Server. O início de sua operação acontece logo após a ativação e estabilização do Cluster e fica disponível enquanto o mesmo estiver operante. Ao identificar uma nova atividade, o programa interpreta as informações e as transforma em uma linha de comando a ser executada via console.

Como exemplo, pode-se citar a operação de BLAST existente no módulo Droid. Abaixo são apresentados os passos envolvidos no processo, também ilustrado pela Figura 17.

1. Usuário realiza o pedido via BAK4BIO Droid;
2. BAK4BIO Server trata o pedido;
3. BAK4BIO Box armazena o arquivo de entrada;
4. BAK4BIO Cluster processa atividade pendente;
5. BAK4BIO Cluster atualiza status do pedido via BAK4BIO Server;
6. BAK4BIO Cluster envia arquivo de saída via BAK4BIO Box;
7. BAK4BIO Server notifica BAK4BIO Droid sobre o resultado;
8. BAK4BIO Droid notifica Usuário.

É importante mencionar que para a validação e demonstração deste módulo foram utilizados apenas processos Blast. Entretanto, o módulo Cluster é capaz de gerenciar, sem necessidade de mudança alguma, a qualquer outra ferramenta de Bioinformática que necessite de processamento paralelo.

3.5 BAK4BIO Droid

Nessa seção é apresentado o módulo responsável pela camada de interação do framework BAK4BIO com os usuários de dispositivos móveis da plataforma Android, o BAK4BIO Droid.

3.5.1 Modelagem

Para a construção de um aplicativo para a plataforma Android, é utilizada uma estrutura de pastas e arquivos pré definidos pela própria plataforma de desenvolvimento, as quais visam garantir a padronização dos projetos, bem como a aplicação do modelo de desenvolvimento MVC. Isso significa ao longo do projeto, um produto com maior qualidade e menos suscetibilidade a falhas.

A Figura 18 apresenta a estrutura inicial de um projeto para a plataforma Android. O diretório `src` contém os fontes responsáveis pelas camadas de controle e modelo do aplicativo. Na sequência encontra-se a pasta `gen`, gerada automaticamente pela IDE e responsável pelos arquivos com as referências dos artefatos do projeto, como: valores de cores, estilos, textos e componentes visuais. Já a pasta `bin` contém os arquivos compilados e a pasta `libs` as bibliotecas externas. Por fim, na pasta `res` é representado a camada de visão e o arquivo `AndroidManifest.xml` informa as configurações gerais do projeto. Completa-se então, o modelo de desenvolvimento MVC.

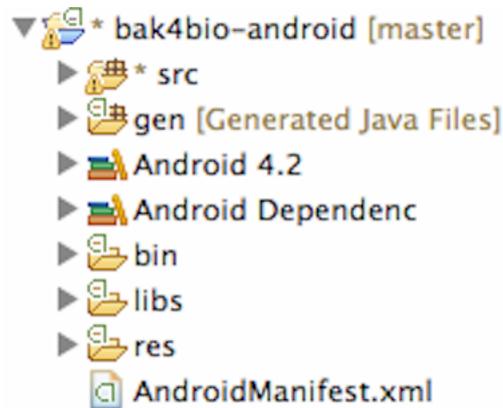


Figura 18: Estrutura inicial de um aplicativo Android.

Fonte: o autor (2013).

3.5.2 Acesso a base de dados biológicas

Funcionalidade com o objetivo de realizar a comunicação e integração com as principais institutos mundiais (NCBI, DDBJ, EBI, PDBJ e KEGG). O processo é realizado com o uso do TogoWS (KATAYAMA; TAKAGI, 2010). Ao receber a requisição, a solução TogoWS processa os parâmetros e conforme o tipo de serviço, é realizado uma nova requisição ao serviço fonte da informação, ou faz uso da sua própria base de dados para prover o resultado. A arquitetura proveniente da comunicação entre o BAK4BIO Droid e o TogoWS é ilustrada na Figura 19.

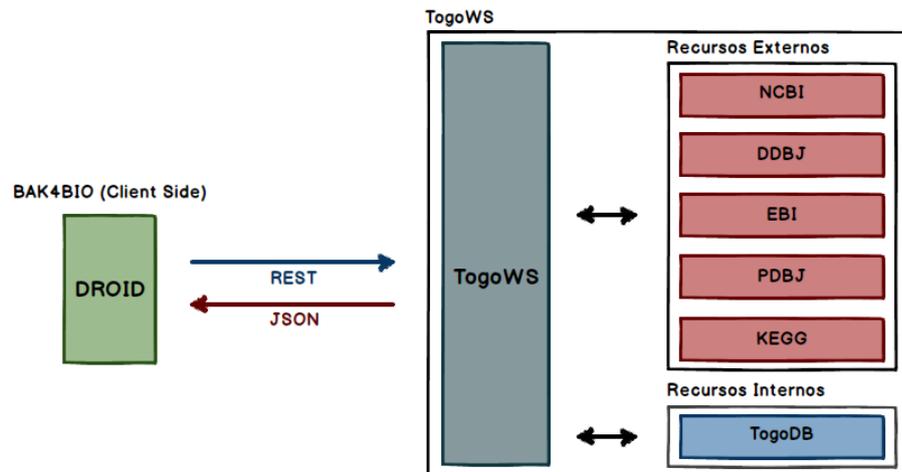


Figura 19: Integração com o TogoWS.

Fonte: o autor (2013).

O fluxo de trabalho se inicia com a execução do aplicativo Android. Ao ser apresentado o menu principal, o usuário é convidado a escolher a instituição pela qual deseja realizar uma pesquisa. Em seguida, é apresentado a relação de bases de dados biológicas da instituição e disponibiliza ao usuário o campo para aplicar o filtro. Ao executar a ação, temos o resultado apresentado e com a opção por maiores detalhes, conforme ilustra a Figura 20.

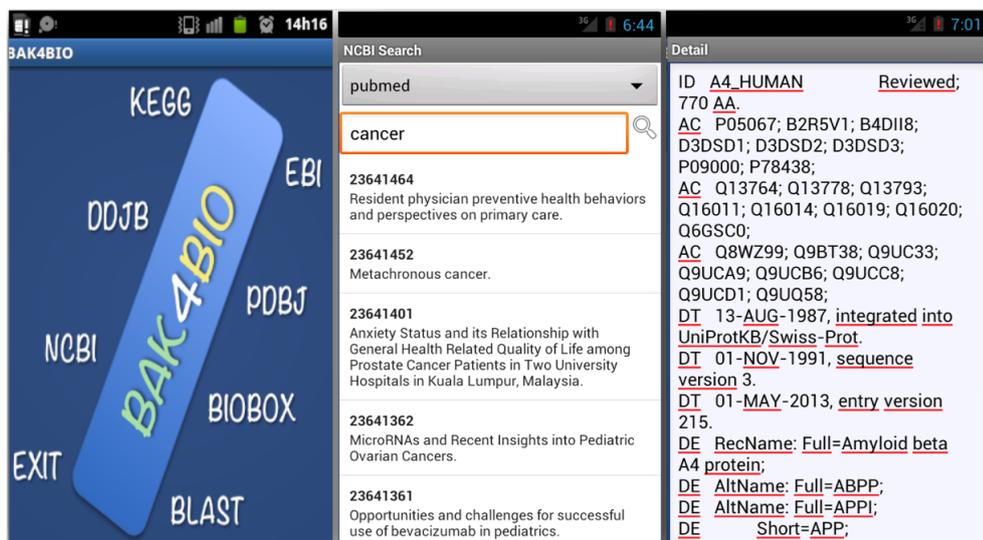


Figura 20: Demonstração do fluxo de trabalho.

Fonte: o autor (2013).

3.5.3 Execução de BLAST

Amplamente utilizado pela comunidade científica em Bioinformática e áreas afins, o BLAST tem como objetivo encontrar regiões de similaridade entre sequências, sendo fundamental para inferir relações funcionais e evolutivas das mesmas, bem como ajudar a identificar os membros das famílias de genes.

Presente em vários procedimentos e na construção de conhecimento, tem papel fundamental no trabalho do bioinformata. Integrado ao módulo BAK4BIO Droid, o objetivo é realizar o procedimento por meio do dispositivo móvel e se tornar uma ferramenta auxiliar no trabalho do pesquisador ou estudante da área.

Conforme apresentado na Figura 21, as interfaces foram projetadas de forma a facilitar o manuseio, tornando o processo mais simples e claro. A funcionalidade exige os seguintes passos:



Figura 21: Interfaces para a realização do BLAST.

Fonte: o autor (2013).

1. Selecionar a opção BLAST no menu;
2. Informar email e senha (cadastrados anteriormente no módulo BAK4BIO Web);
3. Na interface com a lista de operações já realizadas, selecionar a opção novo BLAST;
4. Na tela de parâmetros preencher as informações, conforme descritas na Tabela 2;
5. Com os dados devidamente preenchidos, submeter as informações através da ação "Submit".

Ao receber a requisição, o módulo BAK4BIO Server fica encarregado de armazenar e tornar visível o novo pedido ao módulo administrador do BAK4BIO Cluster, para enfim ser realizado o processamento no Cluster e gerar como saída o novo arquivo no módulo BAK4BIO Box e conseqüentemente atualização do status. Com isso, é possível acompanhar o andamento do trabalho pelo aplicativo Android.

Tabela 2: Parâmetros BLAST

Nome	Descrição
Database	Base de dados
Max target sequence	Núm. máximo de sequências a serem exibidas
Expect threshold	Núm aleatório sobre a expectativa de similaridade
Word size	Comprimento inicial da semente
Max matches range	Limitar o número de execuções
Match/Mismatch Scores	Recompensa e punição por paridade/disparidade
Gap Costs	Custo para criar ou ampliar uma lacuna

Fonte: o autor (2013).

3.5.4 Segurança

O módulo Droid é constituído por duas áreas distintas de acesso: uma pública e outra privada. A primeira é relacionada a integração com os serviços do TogoWS e não requer autenticação para realizar as operações, ou seja, não faz uso de informações de cunho pessoal. A outra se refere a realização de operações BLAST, na qual utiliza-se de dados relacionados ao cadastro do usuário, conforme ilustra a Figura 22.

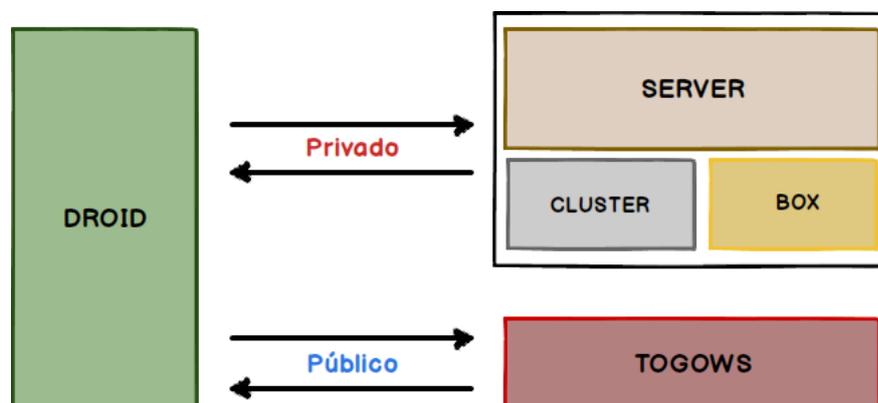


Figura 22: Diferentes áreas de acesso do BAK4BIO Droid.

Fonte: o autor (2013).

3.5.5 Distribuição

Ao realizar a compilação do projeto, ou seja, transformar códigos fontes em arquivos binários capaz de ser interpretado pelos dispositivos móveis, é gerado um arquivo com a extensão APK. Com isso, já é possível realizar a distribuição do aplicativo, seja através de emails, sites ou outros meios de comunicação digital.

A principal forma de distribuição para aplicativos Android é a Google PlayStore, loja virtual mantida pela (GOOGLE INC., 1998). Abaixo são relacionadas algumas vantagens na utilização deste canal:

- Visibilidade mundial do projeto;
- Controle de atualização automática;
- Canal oficial de distribuição;
- Realização de campanhas de marketing;
- Relação entre projeto e autor/instituição produtora da solução.

3.6 BAK4BIO Web

Nessa seção é apresentado o módulo responsável pela camada de interação do *framework* com os usuários através de um navegador, o BAK4BIO Web, responsável em prover o acesso as soluções sem a obrigatoriedade de um aplicativo para as plataformas móveis. Com isso, é possível usufruir da solução em qualquer computador com navegador e acesso à Internet.

3.6.1 Funcionalidades

Abaixo são apresentadas as principais funcionalidades do módulo Web:

- a) Cadastro de usuário;
Possibilita a inclusão de novos usuários para usufruir de funcionalidades com acesso restrito.
- b) Confirmação da conta via email;
Procedimento para garantir a confiabilidade do cadastro, ou seja, garantir que o usuário que realizou o cadastro seja de fato o próprio.

- c) Recuperação de senha via email;
Funcionalidade para usuários já cadastrados e que eventualmente tenham esquecidos da sua senha.
- d) Canal de comunicação com o módulo Box;
Possibilita o gerenciamento e download dos arquivos presentes no BAK4BIO Box.
- d) Canal de comunicação com o módulo Cluster;
Funcionalidade para agendar e acompanhar operações presentes no BAK4BIO Cluster.

3.7 Documentação

Por fim, apresenta-se nos anexos deste documento, a análise e documentação do framework BAK4BIO e seus subsistemas. Esta etapa inicia-se com a apresentação do diagrama de casos de uso, diagramas de classe e finaliza apresentando o MER (Modelo Entidade-Relacionamento).

4 EXPERIMENTOS PRÁTICOS E RESULTADOS

A carência de soluções robustas e completas para o desenvolvimento de ferramentas para Bioinformática encontrada na literatura demonstra que, a existência de um framework preparado para trabalhar com diferentes necessidades, como aplicativos para dispositivos móveis, aplicações para a web, armazenamento de arquivos nas nuvens e capaz de executar processos com alto poder de processamento em uma mesma plataforma, contribui para a integração entre diferentes trabalhos científicos, bem como coloca a solução em destaque por seu pionerismo em âmbito nacional na área de mobilidade.

Este capítulo apresenta os resultados obtidos com o trabalho, com destaque para o módulo BAK4BIO Droid, que já está disponível na Google PlayStore e já vêm sendo utilizado pela comunidade científica. O capítulo apresenta ainda um estudo de caso referente ao acesso às bases de dados biológicas e gerenciamento de um processo de bioinformática no Cluster, o BLAST.

4.1 BAK4BIO Droid

O BAK4BIO Droid teve sua primeira fase de desenvolvimento em meados de setembro de 2012. Nesta versão, o aplicativo fornecia apenas acesso às bases de dados biológicas a partir da integração com o TogoWS. Logo após a finalização desta primeira versão, o mesmo foi cadastrado para download na loja virtual do Google e ainda encontra-se disponível para a comunidade de forma gratuita.

4.1.1 Usuários

Até o momento já foram realizados 287 instalações do aplicativo, desde o seu lançamento em 20/09/2012, conforme Figura 23. É um número expressivo, justamente por se tratar de uma solução bastante específica, com um número restrito de usuários.

Além disto, o aplicativo não passou por uma campanha de marketing profissional.

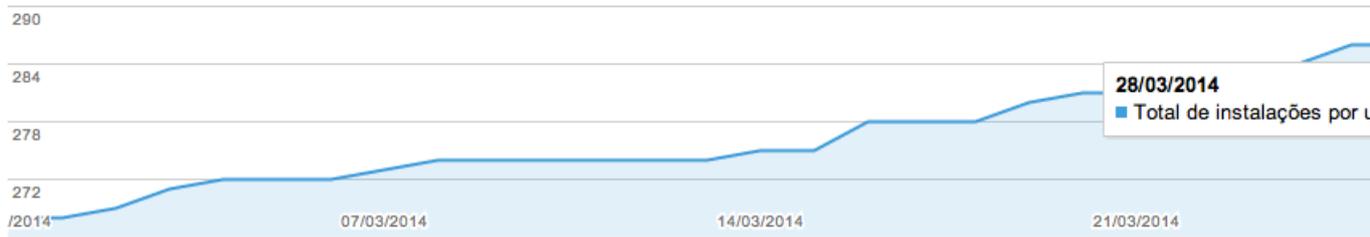


Figura 23: BAK4BIO Droid: Instalações totais.

Fonte: (DEVELOPER CONSOLE GOOGLE PLAY, 2013)

4.1.2 Distribuição geográfica

Desenvolvido no idioma inglês, o aplicativo conseguiu atingir pesquisadores e estudantes da área em vários países, conforme ilustra a Figura 24. A maior concentração está na Índia, Brasil e EUA. Entretanto, pode-se perceber que a categoria Outras representa 39,02% do total, comprovando a efetiva distribuição da solução ao redor do mundo.

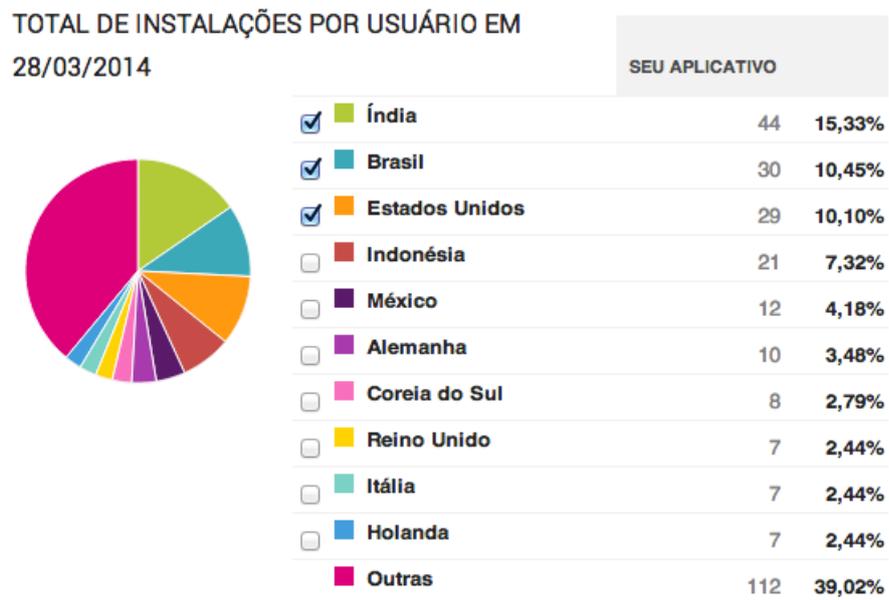


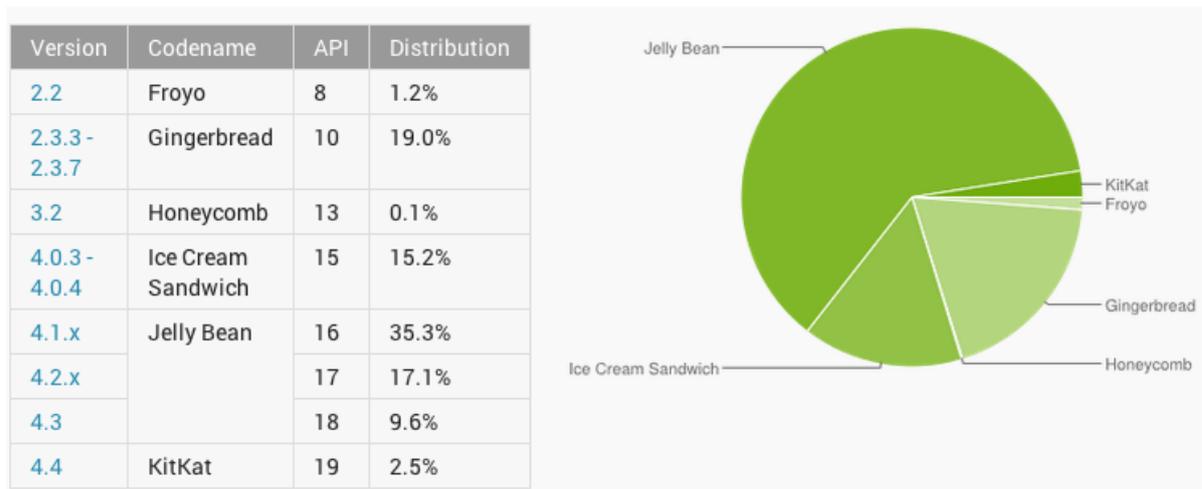
Figura 24: BAK4BIO Droid: Distribuição geográfica.

Fonte: (DEVELOPER CONSOLE GOOGLE PLAY, 2013)

4.1.3 Versões do Android

Construído para executar em *smartphones* e tablets, o aplicativo utilizou como versão mínima o SDK 2.2, representando uma cobertura de 98,2% de compatibilidade com todos dispositivos equipados com o Android e homologados pelo Google até o momento (Figura 25).

Figura 25: Distribuição das versões Android



Fonte: (DEVELOPER CONSOLE GOOGLE PLAY, 2013)

De modo geral, a proporção apresentada na Figura 25 também vale para o aplicativo BAK4BIO Droid (Figura 26).

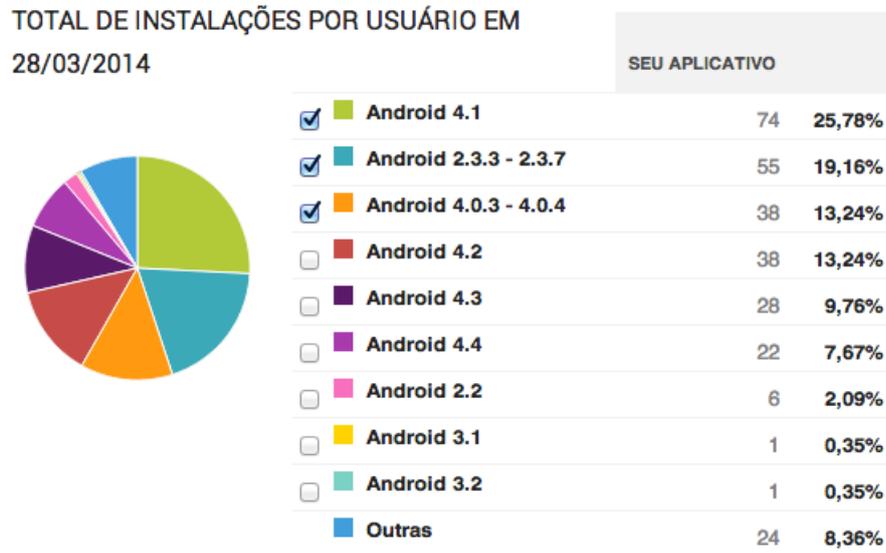
4.2 Estudo de caso

Nessa seção é apresentado um estudo de caso com o objetivo de comprovar a viabilidade de se utilizar o framework BAK4BIO para a construção de soluções mobile e Web para a área de Bioinformática. E para ilustrar um cenário que pode ser amplamente requisitado pelos usuários do BAK4BIO, foi desenvolvida uma experiência com o acesso a base de dados biológicas e execução de *Blast*.

4.2.1 Acesso a base de dados biológicas

Com a intenção de medir o desempenho e comprovar a viabilidade da integração com a API TogoWS, abaixo seguem os resultados obtidos durante a fase de testes. O cenário utilizado para a realização do procedimento é descrito abaixo:

Figura 26: BAK4BIO Droid: Versões do Android



(DEVELOPER CONSOLE GOOGLE PLAY, 2013)

1. Conexão

- (a) Modem D-Link DSL-2730B
- (b) ADSL de 15Mbps (GVT)

2. Cliente (BAK4BIO Droid)

- (a) Tablet Motorola Xoom 2 Media Edition
 - Android 4.0.4 (Ice Cream Sandwich)
 - Display de 8,2 polegadas
 - Processador Dual Core de 1.2GHz
 - Memória RAM de 1GB

O primeiro teste de performance visa garantir a qualidade do serviço na integração com o TogoWS via BAK4BIO Droid com apenas 1 usuário em 10 consultas seguidas (com intervalo de 10 segundos). Como resultado, obteve-se um valor de resposta em média menor que 4 segundos, conforme ilustrado na Tabela 3, considerado um resultado satisfatório de performance.

O segundo teste de performance visa garantir a qualidade do serviço na integração com o TogoWS via BAK4BIO Droid com 10 usuários simultâneos em 10 consultas simultâneas (com intervalos de 10 segundos), desta forma inicia-se a verificação de escalabilidade. Como resultado, manteve-se o valor satisfatório de performance conforme

Tabela 3: Performance integração TogoWS com apenas 1 usuário

Recurso	Bases de dados	Palavra Chave	Tempo Médio (segundos)
NCBI	Pubmed	cancer	2,27
DDBJ	DDBJ	cancer	2,10
EBI	UniProt	cancer	3,72
PDBJ	PDB	cancer	1,6
KEGG	Drug	coca	1,4

ilustrado na Tabela 4, tempo médio da requisição mais demorada ainda menor que 4 segundos. O ótimo resultado se justifica pelo processo ocorrer de forma concorrente, ou seja, cada pedido é tratado por uma thread, em um ambiente multi-thread.

Tabela 4: Performance integração TogoWS com 10 usuários simultâneos

Recurso	Bases de dados	Palavra Chave	Tempo Médio (segundos)
NCBI	Pubmed	cancer	2,56
DDBJ	DDBJ	cancer	2,86
EBI	UniProt	cancer	3,95
PDBJ	PDB	cancer	1,92
KEGG	Drug	coca	1,85

Fonte: o autor (2013).

Por fim, o terceiro teste de performance visa garantir a qualidade do serviço na integração com o TogoWS via BAK4BIO Droid com 50 usuários simultâneos em 10 consultas em sequência (com intervalos de 10 segundos). Como resultado, a performance manteve-se completamente aceitável, tendo em média a última consulta sido respondida ao redor de 5 segundos, como demonstra a Tabela 5. Curiosamente, verificou-se nos testes que a conexão com o banco de dados da EBI é a mais demorada.

Tabela 5: Performance integração TogoWS com 50 usuários simultâneos

Recurso	Bases de dados	Palavra Chave	Tempo Médio (segundos)
NCBI	Pubmed	cancer	4,16
DDBJ	DDBJ	cancer	4,56
EBI	UniProt	cancer	5,17
PDBJ	PDB	cancer	3,32
KEGG	Drug	coca	3,59

4.2.2 Execução de BLAST

Divididos em 4 testes, experimentos práticos foram realizados de forma a medir o desempenho e comprovar a viabilidade de executar operações BLAST. O cenário utilizado para a realização do procedimento é descrito abaixo:

1. Conexão
 - (a) Modem D-Link DSL-2730B
 - (b) ADSL de 15Mbps (GVT)
2. Cliente (BAK4BIO Droid)
 - (a) Tablet Motorola Xoom 2 Media Edition
 - Android 4.0.4 (Ice Cream Sandwich)
 - Display de 8,2 polegadas
 - Processador Dual Core de 1.2GHz
 - Memória RAM de 1GB
3. Servidor Amazon (BAK4BIO Server e Box)
 - (a) Versão Ubuntu 12.10
 - (b) Memória RAM de 613 MiB (MebiByte) ¹
 - (c) Plataforma de 64 bits
 - (d) Processador de 1.8GHz
4. Máquina Laboratório UFPR (BAK4BIO Cloud)
 - (a) Versão Ubuntu 12.10
 - (b) Memória RAM de 4 GB
 - (c) Plataforma de 64 bits
 - (d) Processador DualCore de 2.8GHz

O primeiro teste de performance visa garantir a qualidade do serviço do BAK4BIO Droid para a tarefa de enviar novos pedidos ao módulo Server. Vale mencionar que para este cenário não é preciso levar em consideração o tamanho do arquivo a ser processado,

¹1 MiB = 2²⁰ Bytes.

pois o mesmo já se encontra armazenado e disponível no módulo Box. Como resultado, obteve-se um valor de resposta em média menor que 2 segundos, conforme ilustrado na Tabela 6, considerado um resultado excelente de performance.

Tabela 6: Performance no envió de novas requisições BLAST

Requisições Simultâneas	Tempo Médio (segundos)
10	1,13
20	1,50
30	2,21

Fonte: o autor (2013).

O segundo teste de performance visa garantir a qualidade do serviço na transferência (upload e dowload) do arquivo de entrada entre os módulos Box e Cluster. Foram utilizados conteúdos com tamanho entre 1Mb e 50Mb e obteve-se um valor de resposta em média maior que 30 segundos para arquivos com tamanho aproximado de 20MB, conforme ilustrado na Tabela 7. Conclui-se um resultado regular de performance devido a limitações de processamento existentes nos servidores dos módulos envolvidos.

Tabela 7: Performance na transferência de arquivos de entrada entre os módulos Box e Cluster

Tamanho do Arquivo	Tempo Médio Transferência (segundos)
1,4Mb	3,83
5,7Mb	11,45
18,2Mb	35,21
45,8Mb	132,97

Fonte: o autor (2013).

O terceiro teste de performance visa garantir a qualidade do serviço no processamento da informação no módulo Cluster. Foram utilizados os mesmo arquivos utilizados no segundo teste e obteve-se um valor de resposta em média maior a 2 minutos para arquivos ao redor de 20MB, conforme ilustrado na Tabela 8. Conclui-se um resultado insatisfatório de performance devido a limitações de processamento existentes nos servidores responsável pelo módulo do Cluster.

Por fim, o quarto teste de performance visa garantir a qualidade do serviço na transferência (upload e dowload) do arquivo de saída entre os módulos Cluster e Box. Foram utilizados como base os processamentos dos testes anteriores e obteve-se um valor de resposta em média maior que 25 segundos para arquivos ao redor de 20MB, conforme

Tabela 8: Performance no processamento no módulo Cluster

Tamanho do Arquivo	Tempo Médio Processamento (segundos)
1,4Mb	35,22
5,7Mb	63,48
18,2Mb	193,79
45,8Mb	761,24

Fonte: o autor (2013).

ilustrado na Tabela 9. Conclui-se um resultado insatisfatório de performance devido a limitações de processamento existente no servidor do módulo Cluster.

Tabela 9: Performance na transferência de arquivos de saída entre os módulos Cluster e Box

Tamanho do Arquivo Saída	Tempo Médio Transferência (segundos)
1,1Mb	4,23
3,7Mb	8,82
15,9Mb	26,53
36,8Mb	82,42

Fonte: o autor (2013).

É importante mencionar que os módulos do projeto podem ser ampliados para utilizar outras soluções em Bioinformática. O acesso a base de dados biológicos e o BLAST foram escolhidos para a validação e demonstração do estudo de caso por serem uma das ferramentas mais utilizadas na área e permitir desta forma obter melhores resultados em diferentes estudos e análises.

5 CONCLUSÃO

Construiu-se um framework, denominado BAK4BIO: *Brazilian Army Knife for Bioinformatics* para desenvolvimento de soluções digitais aplicados a área de Bioinformática e afins. O BAK4BIO é composto por cinco módulos, Droid, Web, Server, Box e Cluster; promove uma arquitetura robusta e enxuta, bem como viabiliza a construção e integração de projetos científicos de forma rápida e simples através do uso de tecnologias com alta produtividade e com foco na qualidade.

Responsável pelo canal de comunicação via dispositivos móveis, o BAK4BIO Droid já encontra-se disponível na loja virtual do Google, a PlayStore, desde 20/09/2012 e já somam mais de 250 instalações. Tendo sido utilizado por pesquisadores e estudantes da área em vários países, com destaque para o Brasil, Índia e EUA. Por permitir a consulta aos principais bancos de dados biológicos (NCBI, DDBJ, EBI, PDBJ e KEGG) e operações de BLAST, o BAK4BIO Droid centraliza e facilita os trabalhos diários de um Bioinformata.

Seguindo o framework, o módulo BAK4BIO Server é encarregado pelo gerenciamento das etapas de autorização e autenticação, bem como assume papel fundamental na comunicação entre módulos do framework. É o módulo principal para realizar novas integrações. É este módulo que recebe todas as requisições de usuários e as envia para o destino correto, sendo uma base de dados biológica, um cluster de processamento paralelo ou o BAK4BIO Box.

Com o objetivo de armazenar arquivos provenientes de entradas e saídas dos processos envolvidos na solução, o módulo BAK4BIO Box foi construído com base na computação em nuvem, especificamente nos conceitos de IaaS (*Infrastructure as a Service*). Este módulo fornece acesso aos arquivos dos usuários de forma centralizada e sem a necessidade de instalação de softwares auxiliares.

O módulo BAK4BIO Web representa o canal de comunicação via Web, responsável pelo gerenciamento de novos usuários, bem como a manutenção dos arquivos

presentes no módulo Droid e operações consumidas pelo módulo Cluster.

Por fim, responsável pela execução e paralelismo de operações com demandas de alto poder computacional, o módulo BAK4BIO Cluster assume a função de consumidor de tarefas produzidas pelos usuários finais, as quais são iniciadas a partir dos módulos Web ou Droid.

A solução construída demonstrou um excelente desempenho, com um bom grau de escalabilidade. Resultados obtidos mostram que o comportamento do framework não é afetado conforme o número de usuários simultâneos utiliza seus subsistemas. A solução obteve um tempo médio de resposta ao redor de 5 segundos quando testada para requisições de até 50 usuários simultaneamente, demonstrando um bom grau de escalabilidade.

Pelo fato de viabilizar a construção do framework, estabelecer uma ponte entre a comunidade global e Programa de Pós Graduação em Bioinformática da UFPR, por meio de um aplicativo presente em vários países, bem como tornar o mesmo pioneiro no desenvolvimento de aplicativos móveis em Bioinformática no Brasil, torna-se legítima a afirmação de que este trabalho contribui para o crescimento do Programa, torna acessível novas tecnologias e padroniza a forma de construir novos projetos por meio de um framework com as mais recentes tecnologias do mercado.

6 TRABALHOS FUTUROS

Este capítulo finaliza a dissertação apresentando as considerações para trabalhos futuros. Este encontra-se dividido em seções que representam melhorias para cada um dos módulos do trabalho.

6.1 Geral

Site explicativo sobre o projeto, as funcionalidades, bem como os objetivos e documentação.

6.2 BAK4BIO Droid

Melhorias na interface de resultado para alguns serviços, como NCBI e KEGG.

Cadastro de Usuário via dispositivo móvel.

Interface para integração com outros projetos do Programa de Pós Graduação em Bioinformática da UFPR ou parceiros.

6.3 BAK4BIO Web

Melhorias nas interfaces envolvidas na integração com a biblioteca Devise.

Possibilitar o acesso aos serviços do TogoWS no módulo.

6.4 BAK4BIO Box

Compartilhamento de arquivos entre usuários.

Manutenção de arquivos com agrupamento (diretórios).

Integração com os serviços do DropBox.

6.5 BAK4BIO Server

Ser a ponte entre o módulo Droid e os serviços disponíveis no TogoWS, tirando esta tarefa do módulo BAK4BIO Droid.

6.6 BAK4BIO Cluster

Aumentar o número de bancos de dados suportados na operação de BLAST.

Considerar todos os parâmetros extras na execução do BLAST.

Inclusão de outras ferramentas de Bioinformática.

Integração com outros projetos do Programa de Pós Graduação em Bioinformática da UFPR ou parceiros.

Inclusão de múltiplos clusters de processamento e um módulo de distribuição de tarefas para estes clusters de forma equilibrada. Misturando no módulo características de alto desempenho e balanceamento de carga.

Referências

- A VALENTIN F, A. M. L.; R, L. Web services at the european bioinformatics institute. *Nucleic Acids Research*, v. 35, 2007.
- AMERICAN, S. *Understanding Supercomputing*. [S.l.]: Grand Central Publishing, 2002.
- ANDROID DEV. *Publish Checklist for Google Play*. 2013. Disponível em: <<http://developer.android.com/distribute/googleplay/publish/preparing.html>>.
- A.T VELTE T.J, E. R. V. *Cloud Computing, A Practical Approach*. [S.l.]: Paperback, 2009.
- BENSON ILENE KARSCH-MIZRACHI, e. a. D. A. Genbank. *Nucleic Acids Res.* 2005, n. 33, 2005.
- BORGES, V. C. M.; DANTAS, M. Utilização de dispositivos móveis e gerenciamento de workflow para submissão e monitoração de tarefas em ambientes de grids. *Sessão de Pôsteres dos Anais da Escola Regional de Alto Desempenho*, 2006.
- BUYYA, C. S. R. *Cluster Computing*. [S.l.]: Nova Science Pub Inc., 2001.
- BUYYA, R. *High Performance Cluster Computing: Architectures and Systems*. [S.l.]: Prentice Hall, 1999.
- CANCAN. 2009. Disponível em: <<https://github.com/ryanb/cancan>>.
- CROCKFORD, D. *JavaScript: The Good Parts*. [S.l.]: O'Reilly Media, 2008.
- DEVELOPER CONSOLE GOOGLE PLAY. *Developer Console - Google Play*. 2013. Disponível em: <<https://play.google.com/apps/publish/>>.
- EW BARRETT T, e. a. S. Database resources of the national center for biotechnology information. *Nucleic Acids Research*, v. 37, 2009.
- FIELDING, R. T. *Architectural Styles and the Design of Network-based Software Architectures*. Tese (Doutorado) — University of California, Irvine, 2000.
- FLUHR, Z.; NUSSBAUM, E. Switching plan for a cellular mobile telephone system. *IEEE Transactions on Communications volume 21*, 1973.
- GAMMA R. HELM, e. a. E. *Design Patterns Elements of Reusable Object-Oriented Software*. [S.l.]: Addison-wesley Professional, 1998.
- GOOGLE INC. *Google*. 1998. Disponível em: <<http://www.google.com>>.
- H, S. Biological soap servers and web services provided by the public sequence data bank. *Nucleic Acids Research*, v. 31, 2003.

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

HAU, M. M. J.; PAGNI, M. Hitkeeper, a generic software package for hit list management. *Source Code for Biology and Medicine*, v. 2, n. 2, 2007.

JAMES M. M. RAO, e. a. R. A. The hematopoietic expression viewer: expanding mobile apps as a scientific tool. *Oxford Bioinformatics*, v. 28, n. 14, 2012.

JSON. 1999. Disponível em: <<http://www.json.org/>>.

KAMINUMA TAKEHIDE KOSUGE, e. a. E. Ddbj progress report. *Nucleic. Acids Res. 2011*, n. 39, 2010.

KATAYAMA, M. N. T.; TAKAGI, T. Togows: integrated soap and rest apis for interoperable bioinformatics web services. *Nucleic Acids Research*, v. 38, 2010.

KUMAR K. BOCCIA, M. M. S.; YE, J. Exploring spatial patterns of gene expression from fruit fly embryogenesis on the iphone. *Oxford Bioinformatics*, v. 28, n. 21, 2012.

LECHETA, R. *Google Android*. Terceira. [S.l.]: Novatec, 2013.

LEINONEN* RUTH AKHTAR, e. a. R. The european nucleotide archive. *Nucleic. Acids Res. 2011*, n. 39, 2010.

M GOTO S, e. a. K. Kegg for representation and analysis of molecular networks involving diseases and drugs. *Nucleic Acids Research*, v. 38, 2010.

MOTOROLA. *About Motorola*. 2013. Disponível em: <http://www.motorola.com/us/consumers/about-motorola-us/About_Motorola_-_History_-_Timeline/About_Motorola_-_History_-_Timeline.html>.

MOTOROLA ON DISPLAY. 2012. Disponível em: <<http://www.youtube.com/watch?v=0VklpT3-kBM>>.

OLIVEIRA E. OGASAWARA, e. a. D. An adaptive parallel execution strategy for cloud-based scientific workflows. *Concurrency and Computation: Practice and Experience*, v. 24, n. 13, 2012.

PACHECO, R. O. C. M. A. Um estudo de caso da plataforma android com interfaces adaptativas. *Revista @Lumni*, v. 1, n. 1, 2011.

PAGNI, J. H. M.; STOCKINGER, H. A multi-protocol bioinformatics web service: Use soap, take a rest or go with html. *Proc. of the IEEE Int. Symp. on Cluster Computing and the Grid*, 2008.

PDB. *Protein Data Bank*. 2006. Disponível em: <<http://www.rcsb.org/pdb/home/home.do>>.

PINTAS D. OLIVEIRA, e. a. J. Monitoramento em tempo real de workflows científicos executados em paralelo em ambientes distribuídos. *Anais do Congresso da Sociedade Brasileira de Computação*, 2012.

- PLATAFORMATEC. *Devise*. 2013. Disponível em: <<https://github.com/plataformatec/devise>>.
- REUTERS. *Crescimento do mercado de smartphones anima operadoras*. 2012. Disponível em: <<http://info.abril.com.br/noticias/mercado/crescimento-do-mercado-de-smartphones-anima-operadoras-13122012-36.shl>>.
- RICHARDSON, S. R. B. L. *RESTful Web Services*. [S.l.]: O'Reilly Media, 2007.
- RICHARDSON, S. R. L. *RESTful Web Services*. [S.l.]: O'Reilly Media, 2007.
- RIIKONEN J. BOBERG, T. S. P.; VIHINEN, M. Biowap, mobile internet service for bioinformatics. *Oxford Bioinformatics*, v. 17, n. 9, 2001.
- RUBY ON RAILS. *Ruby on Rails*. 2005. Disponível em: <<http://rubyonrails.org/>>.
- RUBY TOOLBOX. *Ruby Toolbox*. 2013. Disponível em: <<http://www.ruby-toolbox.com>>.
- RUIC, G. Android já tem 72% do mercado de smartfones. v. 11, 2012.
- S. THOMAS D., H. D. R. *Agile Web Development with Rails*. [S.l.]: The Pragmatic Programmers, 2011.
- SADEK, H. A. *Bioinformatics: Principles and Basic Internet Applications*. [S.l.]: Trafford Publishing, 2004.
- SAGER, I. *Before iPhone and Android Came Simon, the First Smartphone*. 2012. Disponível em: <<http://www.businessweek.com/articles/2012-06-29/before-iphone-and-android-came-simon-the-first-smartphone>>.
- SCOOP, P. 2008. Disponível em: <<http://www.phonescoop.com/glossary/term.php?gid=131>>.
- SHARGI, H. A. Biological information processing as cloud computing. *Proc. of the IEEE Int. Conf. Applications of Digital Information and Web Technologies*, 2009.
- SIMALIGN. 2012. Disponível em: <<https://itunes.apple.com/us/app/simalign/id432818873?mt=8>>.
- SIMGENE. 2011. Disponível em: <<https://itunes.apple.com/us/app/simgene/id427772349?mt=8>>.
- SNELL D. TIDWELL, P. K. J. *Programming Web Services with SOAP*. [S.l.]: O'Reilly Media, 2001.
- TAMAE, P. R. L. R. Y. Web sservices: Uma nova visão da arquitetura de aplicações distribuídas na internet. *Revista Científica Eletrônica de Sistemas de Informação*, v. 1, n. 2, 2005.
- TARGET TRUST. *O que é cloud computing?* 2009. Disponível em: <<http://www.targettrust.com.br/>>.

TEAM, G. *Mobile phone evolution: Story of shapes and sizes*. 2010. Disponível em: <http://www.gsmarena.com/mobile_phone_evolution_review_493p1.php>.

TYAGI, S. *RESTful Web Services*. 2006. Disponível em: <<http://www.oracle.com/technetwork/articles/javase/index-137171.html>>.

UNIPROT. *Universal Protein Resource*. 2008. Disponível em: <<http://www.uniprot.org>>.

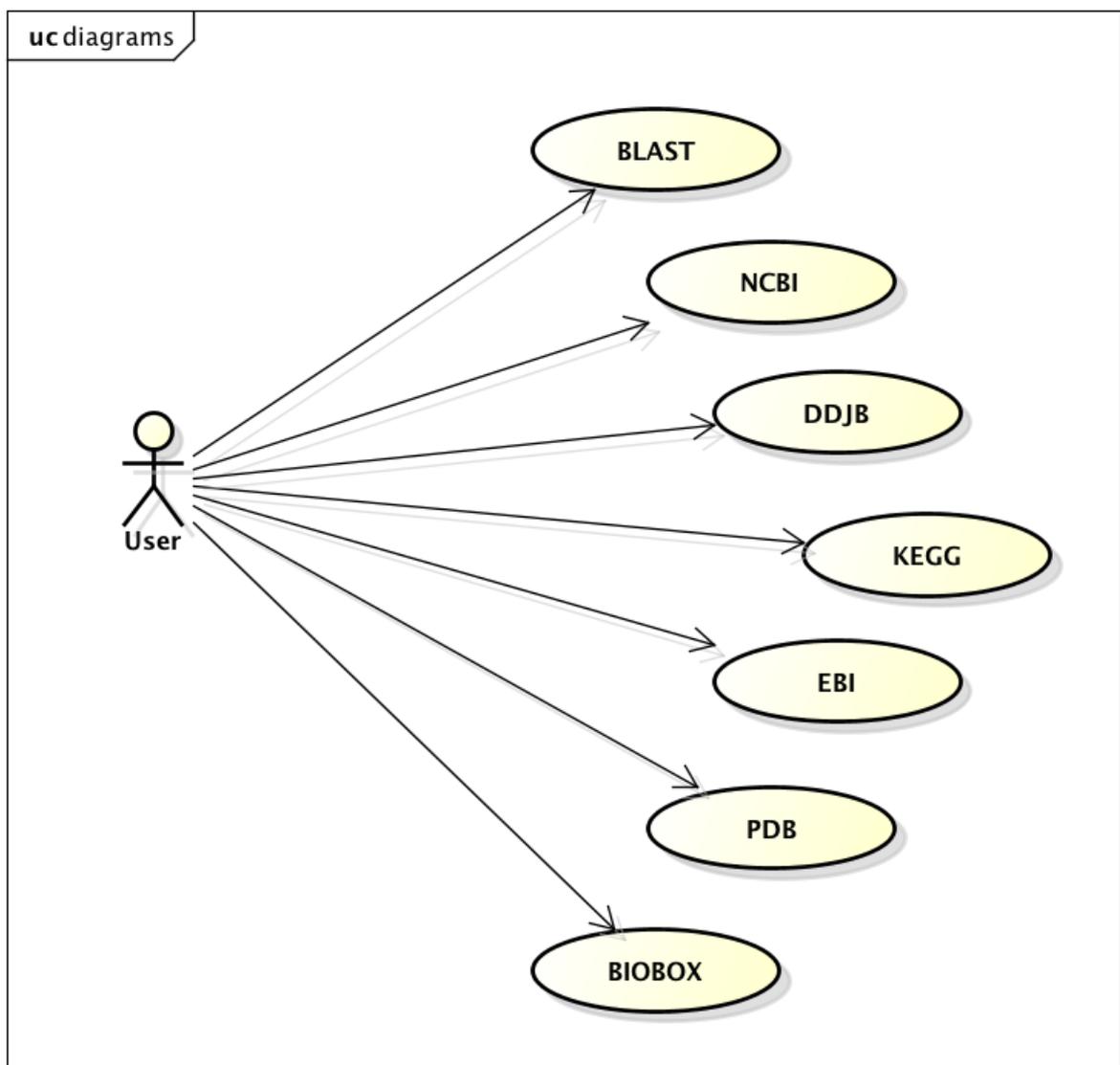
W3C. *Web Services Activity*. 2002. Disponível em: <<http://www.w3.org/2002/ws/>>.

W3C. *Web Services Architecture*. 2004. Disponível em: <<http://www.w3.org/TR/ws-arch/>>.

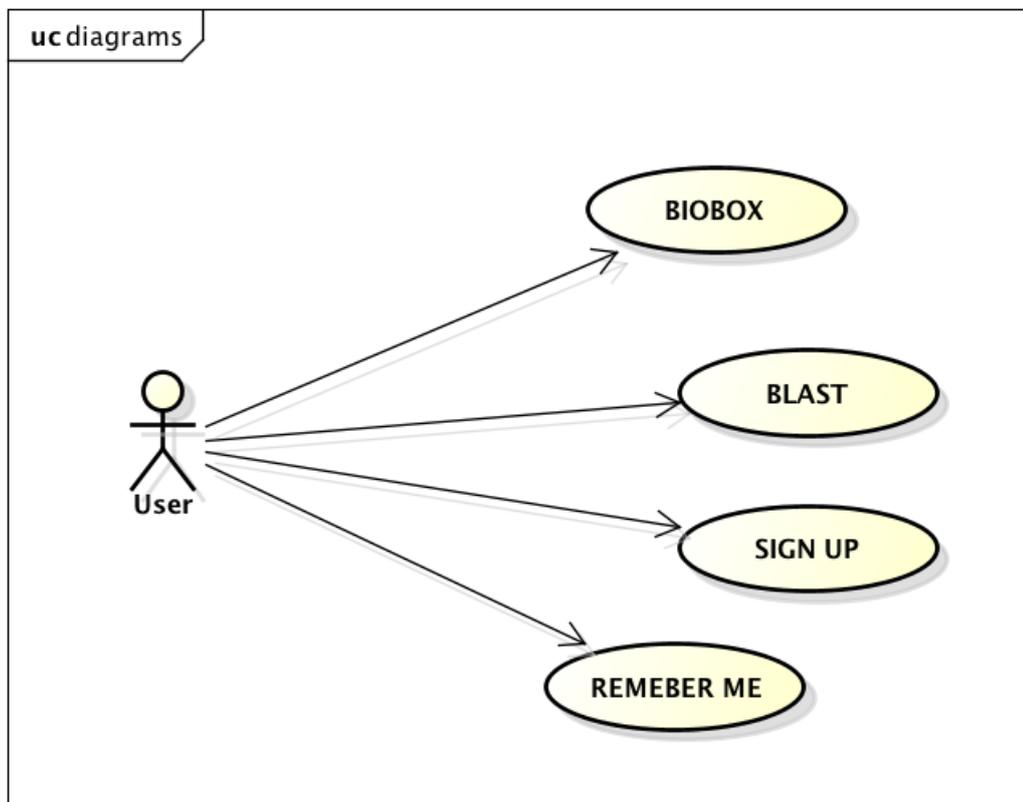
WANG, . A. M. Z.; KISSINGER, J. C. Ws-biozard: A wizard for composing bioinformatics web services. *Proc. of the IEEE Int. Congress on Services*, 2008.

YU X. BAI, e. a. H. Metainformation and workflow management for solving complex problems in grid environments. *Proc. of the IEEE Int. Parallel and Distributed Processing Symp.*, 2004.

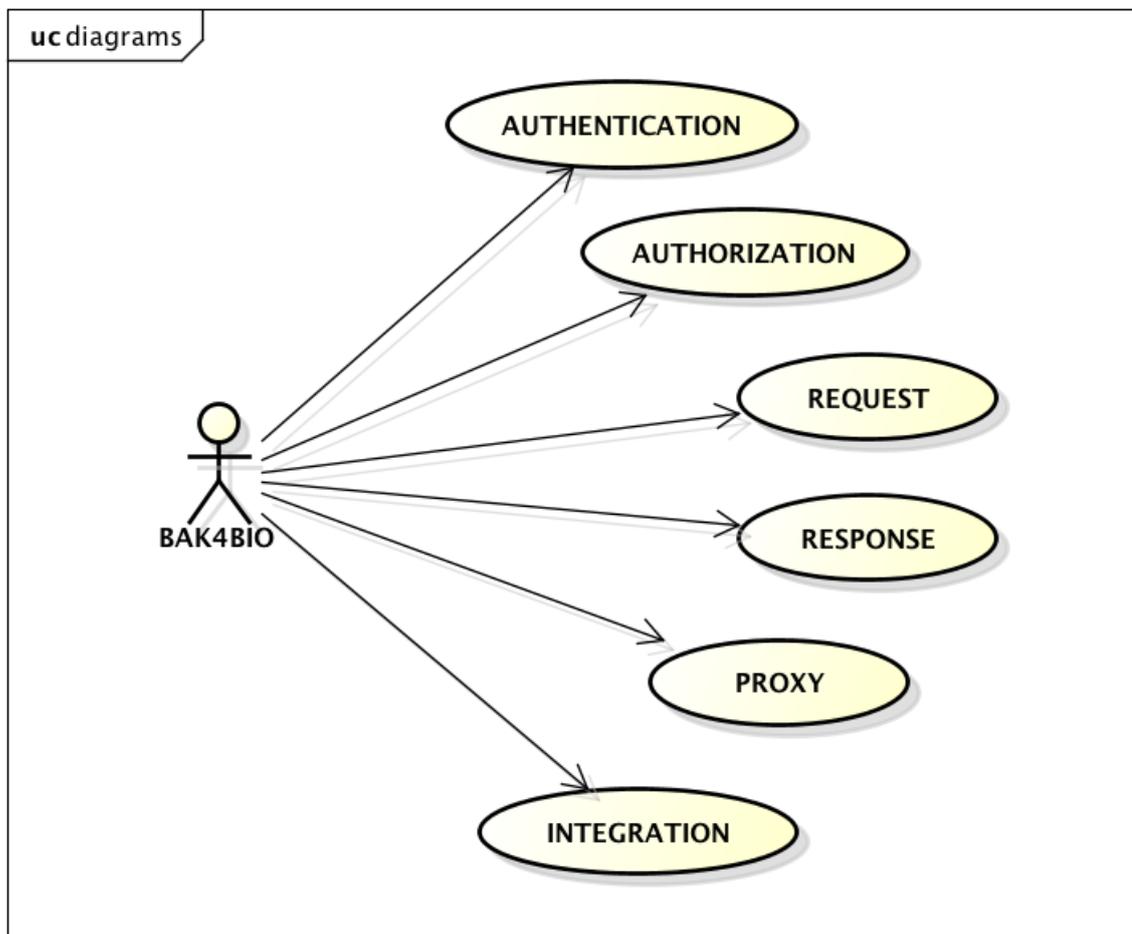
ANEXO A - BAK4BIO Droid: Caso de Uso



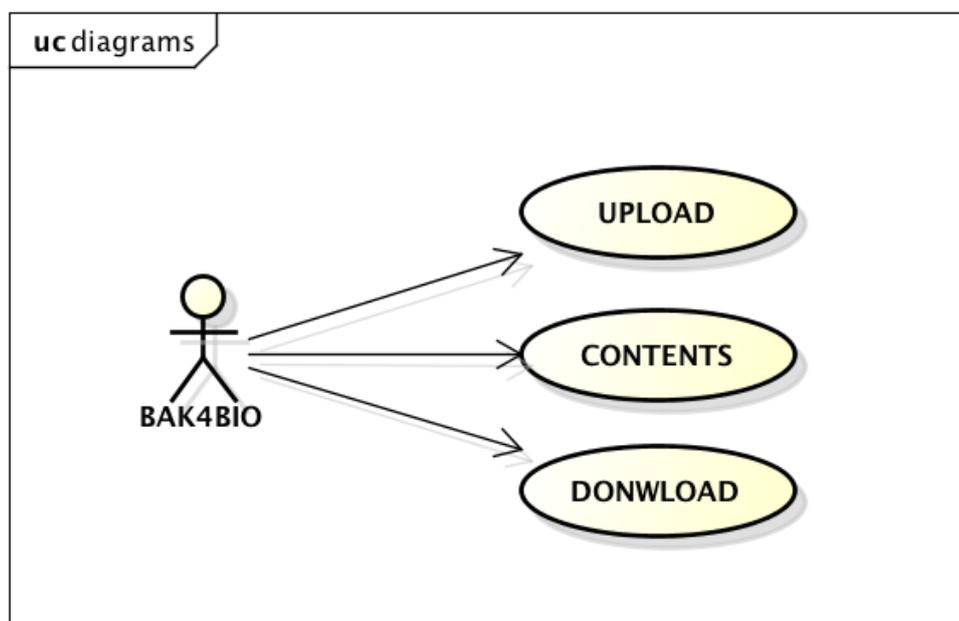
ANEXO B - BAK4BIO Web: Caso de Uso



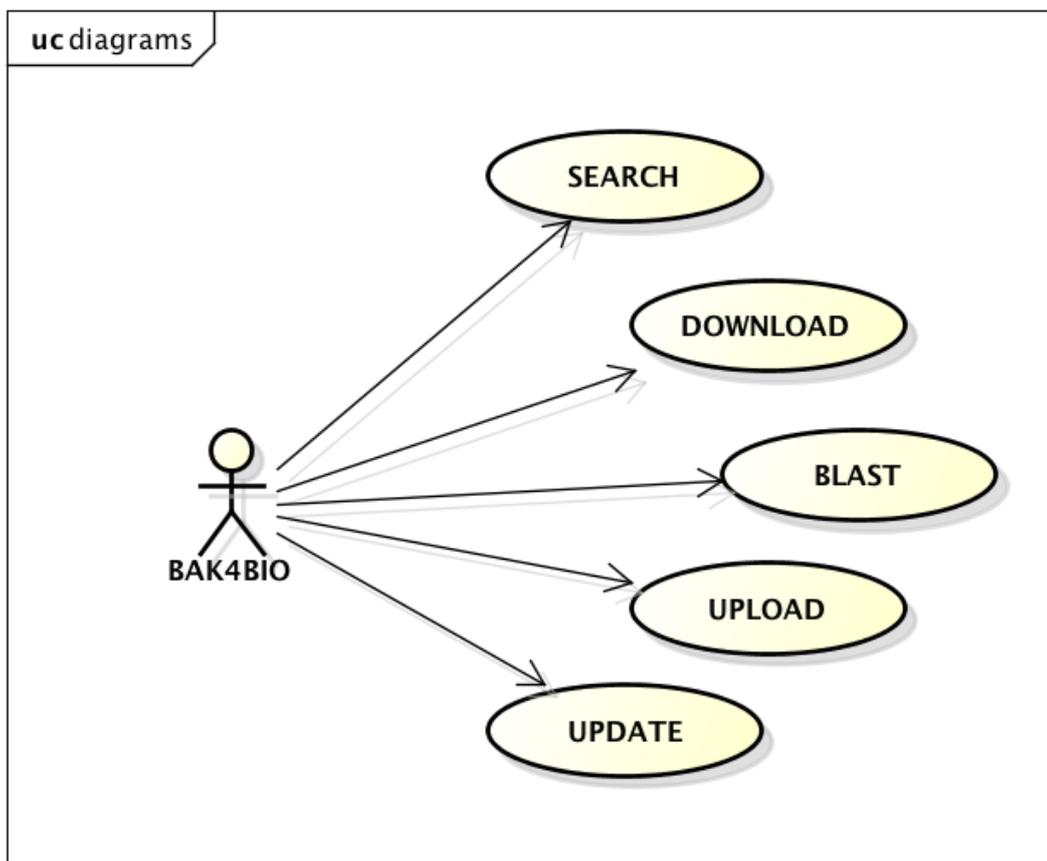
ANEXO C - BAK4BIO Server: Caso de Uso



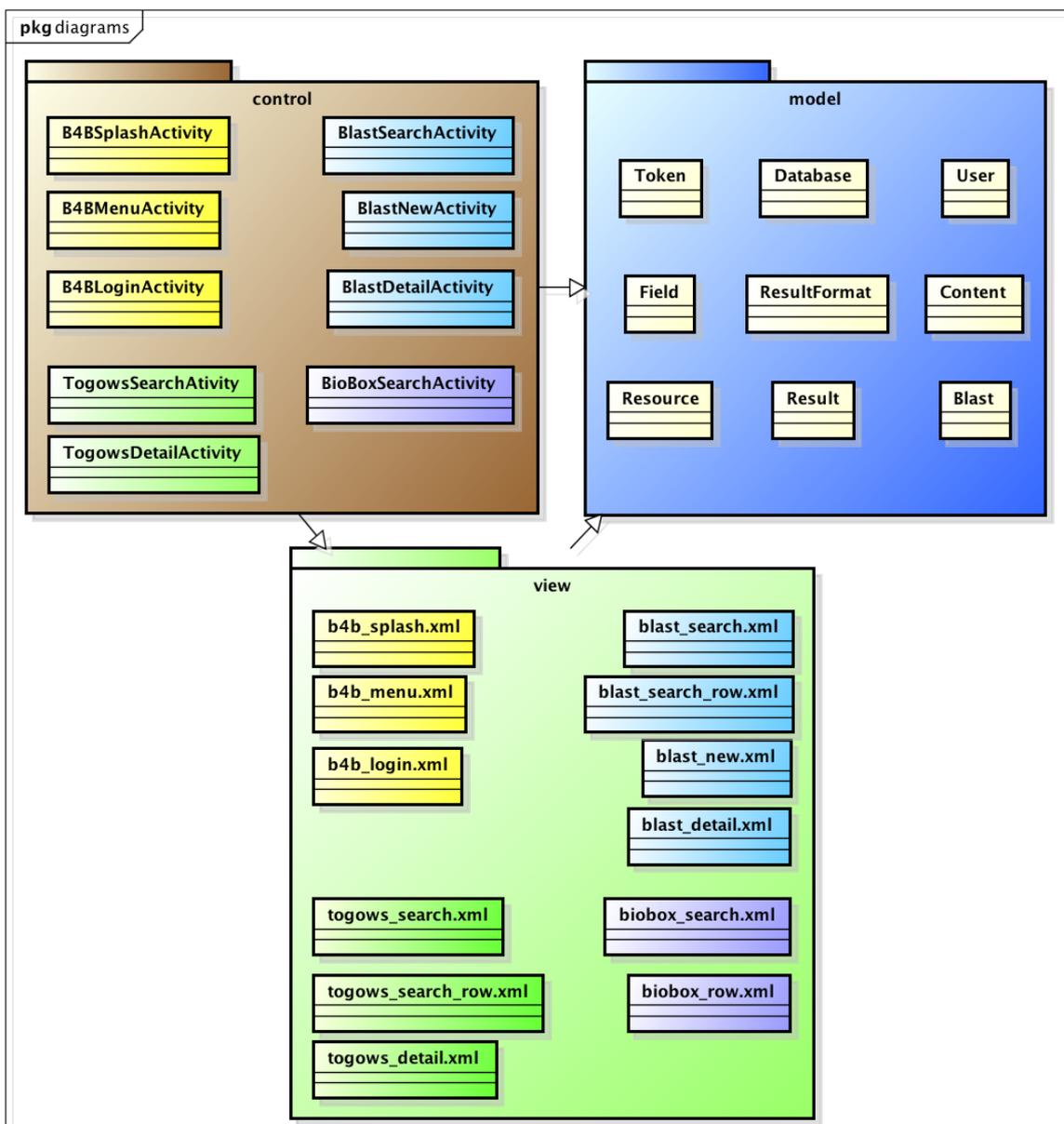
ANEXO D - BAK4BIO Box: Caso de Uso



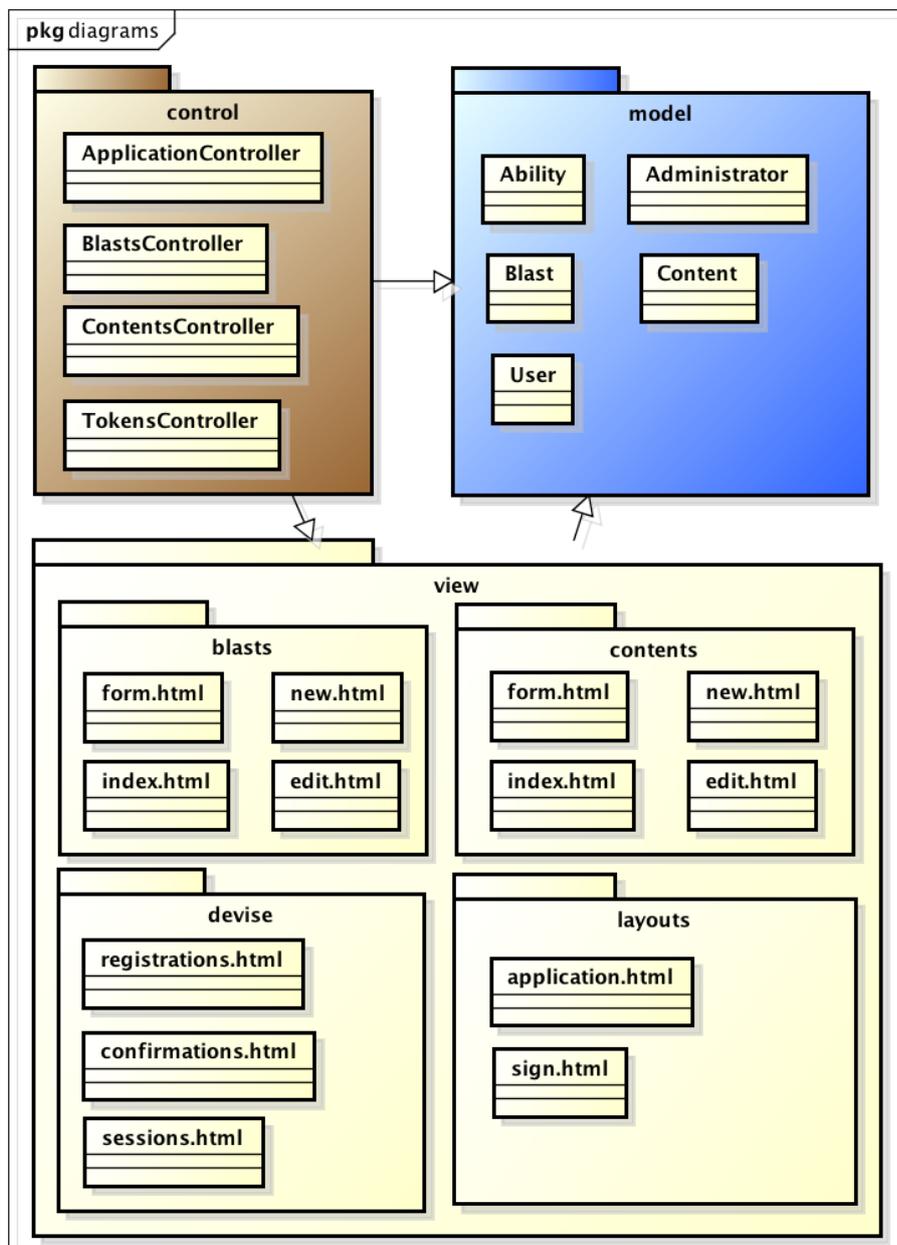
ANEXO E - BAK4BIO Cluster: Caso de Uso



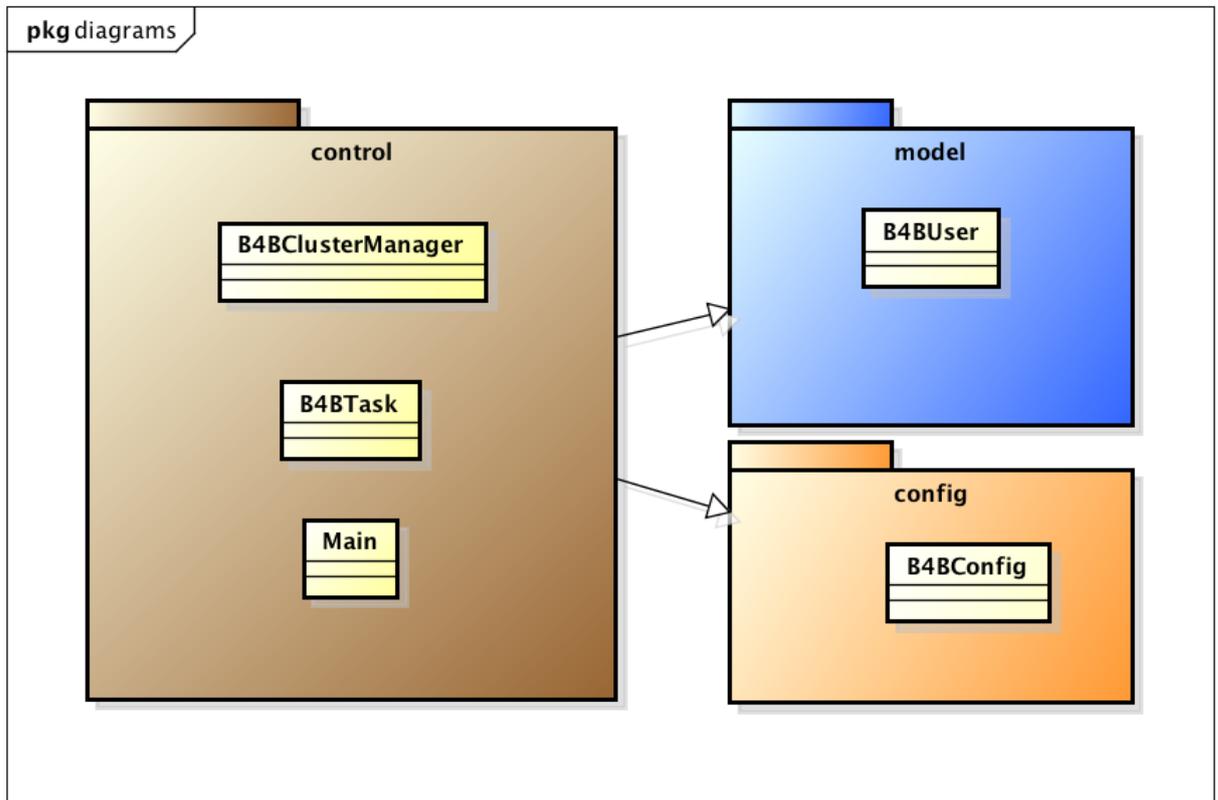
ANEXO F - BAK4BIO Droid: Diagrama de Classes



ANEXO G - BAK4BIO Server: Diagrama de Classes



ANEXO H - BAK4BIO Cluster: Diagrama de Classes



ANEXO I – BAK4BIO: Modelagem Banco de Dados

