UNIVERSIDADE FEDERAL DO PARANÁ

ALLAN COSTA VIEIRA

FERNANDO DE OLIVIEIRA MELO

BASE DE DADOS - FUNGOS PATOLÓGICOS

CURITIBA

2016

ALLAN COSTA VIEIRA

FERNANDO DE OLIVIEIRA MELO

BASE DE DADOS - FUNGOS PATOLÓGICOS

Trabalho de conclusão de curso apresentado como requisito à conclusão do curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Setor de Educação Profissional e Tecnologia, Universidade Federal do Paraná.

Orientador: Jeroniza Nunes Marchaukoski

CURITIBA

2016

**RESUMO**

Arquivos GFF e GBK guardam as mais diversas informações genéticas de um organismo, sendo aplamente utilizados na Bioinformática, mas apesar de possuírem a mesma informação, cada formato as armazena de forma diferente. Este trabalho apresenta um sistema web capaz de guardar e converter os arquivos entre esses dois formatos, assim como disponibiliza-los para download. O sistema também funcionará como uma plataforma de trabalho, facilitando o armazenamento e vizualização de arquivos genéticos e possibilitando que o andamento dos projetos seja facilmente consultado, arquivos disponibilizados e usuários gerenciados de acordo com uma hierarquia estabelecida.

Palavra Chave: Bioinformática, Arquivos genéticos, Visualização de dados biológicos, GFF, GBK

**ABSTRACT**

GFF and GBK files store the most diverse genetic information of an organism, being widely used in bioinformatics, but despite having the same information, each format has different ways of storing. This paper presents a web system capable of store and convert files between these formats, and offers them for download. The system will also function as a work platform, facilitating the storage and visualization of genetic files and enabling the progress of each project to be easily consulted, files made available and users managed in accordance with an established hierarchy.

Key-word: Bioinformatics, Genetic files, Visualization of biological data, GFF, GBK

**LISTA DE FIGURAS**

FIGURA 1 – ARQUIVO FASTA.................................................................................14

FIGURA 2 – DIAGRAMA WBS..................................................................................21

FIGURA 3 – DIAGRAMA DE GANTT........................................................................22

FIGURA 4 – LINHA DO TEMPO DO DIAGRAMA DE GANTT..................................23

FIGURA 5 – LINHA DO TEMPO DO DIAGRAMA DE GANTT..................................23

FIGURA 6 – FOTO TIRADA NO NÍVEL 1 DE COMPRESSÃO.................................33

FIGURA 7 – FOTO TIRADA NO NÍVEL 5 DE COMPRESSÃO.................................34

FIGURA 8 – COMPARAÇÃO ENTRE A FOTO ORIGINAL E A IMAGEM APÓS O PROCESSAMENTO DA IMAGEM.......................................................36

FIGURA 9 - TELA DE ABERTURA............................................................................37

FIGURA 10 - TELA INICIAL DO APLICATIVO..........................................................38

FIGURA 11 - TELA DA CÂMERA..............................................................................38

FIGURA 12 - TELA PARA SELECIONAR UMA IMAGEM.........................................39

FIGURA 13 - TELA DE RESULTADOS.....................................................................40

FIGURA 14 - TELA DE RESULTADOS COM HSPS.................................................40

FIGURA 15 - IMAGEM ORIGINAL NA ABA “DEBUG”..............................................41

FIGURA 16 - IMAGEM ORIGINAL PRÉ-PROCESSADA NA ABA “DEBUG”...........42

FIGURA 17 - SEQUÊNCIA ACHADA PELO OCR NA ABA “DEBUG”......................42

FIGURA 18 - BOTÃO PARA LIMPAR OS RESULTADOS........................................43

FIGURA 19 - ALERTA DE CONFIRMAÇÃO..............................................................44

FIGURA 20 - MENU NA TELA INICIAL DO APLICATIVO.........................................45

FIGURA 21 - TELA DE INFORMAÇÕES GERAIS....................................................46

FIGURA 22 - TELA DE INFORMAÇÕES GERAIS....................................................46

FIGURA 23 - TELA DE CONFIGURAÇÕES..............................................................47

FIGURA 24 - TELA DO APLICATIVO NA PLAY STORE...........................................48

FIGURA 25 – DIAGRAMA DE CASOS DE USO.......................................................54

FIGURA 26 – DIAGRAMA ENTIDADE RELACIONAMENTO....................................62

FIGURA 27 – DIAGRAMA DE CLASSES – PACOTE ACTIVITIES...........................63

FIGURA 28 – DIAGRAMA DE CLASSES – PACOTE DATA.....................................64

FIGURA 29 – DIAGRAMA DE CLASSES – PACOTE PROCESSING......................65

FIGURA 30 – DIAGRAMA DE CLASSES – PACOTE SYSTEM................................66

FIGURA 31 – DIAGRAMA DE ATIVIDADES – USO GERAL....................................67

FIGURA 32 – DIAGRAMA DE ATIVIDADES – ALTERAR CONFIGURAÇÕES........68

FIGURA 33 – DIAGRAMA DE COMPONENTES……………………………………….69

**LISTA DE SIGLAS**

DNA -Ácido desoxirribonucleico

FASTA -Formato utilizado para armazenar sequências de bases e de

aminoácidos em arquivo texto

GenBank -Banco de dados público do National Center for Biology Information

GBK -GenBank File

GFF -General Feature Format

HTML -HyperText Markup Language

IDE -Integrated Development Environment

MVC -Model-View-Controller

NCBI -National Center for Biotechnology Information

NR -Banco de dados não redundante disponibilizado pelo NCBI

PPG-BIOINFO - Programa de Pós-graduação em Bioinformática

RNA -Ácido ribonucleico

RUP -Rational Unified Process

SGBD -Sistema Gerenciador de Bancos de Dados

UML -Unified Modeling Language

WBS -Work Breakdown Structure

XML -Extensible Markup Language

**LISTA DE TABELAS**

TABELA 1 – PLANO DE RISCOS..............................................................................24

SUMÁRIO

[1 INTRODUÇÃO 11](#_Toc405835694)

[1.1 OBJETIVOS 12](#_Toc405835695)

[1.2 OBJETIVOS ESPECÍFICOS 12](#_Toc405835696)

[2 FUNDAMENTAÇÃO TEÓRICA 14](#_Toc405835697)

[2.1 BIOINFORMÁTICA 14](#_Toc405835698)

[2.2 CONCEITOS BIOLÓGICOS 14](#_Toc405835699)

[2.3 CONSULTA DE INFORMAÇÕES BIOLÓGICAS 15](#_Toc405835700)

[2.3.1 Armazenamento 15](#_Toc405835701)

[2.3.2 Banco de Dados 15](#_Toc405835702)

[2.3.3 BLAST 16](#_Toc405835703)

[2.4 OCR 17](#_Toc405835704)

[2.5 PROCESSAMENTO DE IMAGENS 18](#_Toc405835705)

[2.6 SQLITE 18](#_Toc405835706)

[2.8 UML 19](#_Toc405835707)

[3 METODOLOGIA 20](#_Toc405835708)

[3.1 MODELO DE PROCESSO DE ENGENHARIA DE SOFTWARE 20](#_Toc405835709)

[3.2 PLANO DE ATIVIDADES 20](#_Toc405835710)

[3.2.1 Diagrama WBS 20](#_Toc405835711)

[3.2.2 Diagrama de Gantt 23](#_Toc405835712)

[3.3 PLANO DE RISCOS 25](#_Toc405835713)

[3.4 MATERIAIS 26](#_Toc405835714)

[3.4.1 Ambiente de Desenvolvimento 26](#_Toc405835715)

[3.4.2 Linguagem de Programação 26](#_Toc405835716)

[3.4.3 Banco de Dados 27](#_Toc405835717)

[3.4.4 Ambiente Integrado de Desenvolvimento 27](#_Toc405835718)

[3.4.5 Ferramenta para Criação do Diagrama WBS 27](#_Toc405835719)

[3.4.6 Ferramenta para Criação do Diagrama de Gantt 27](#_Toc405835720)

[3.4.7 Ferramenta para Modelagem UML 28](#_Toc405835721)

[3.4.8 Ferramenta para Criação do Diagrama Entidade Relacionamento 28](#_Toc405835722)

[3.5 FERRAMENTA PARA CONTROLE DE VERSÃO 28](#_Toc405835723)

[3.6 BIBLIOTECAS 28](#_Toc405835724)

[3.6.1 BioJava 29](#_Toc405835725)

[3.6.2 Tess-two 29](#_Toc405835726)

[4 DESENVOLVIMENTO DO PROJETO 30](#_Toc405835727)

[4.1 VALIDAÇÃO DO SOFTWARE 31](#_Toc405835728)

[4.1.1 Testes de OCR 31](#_Toc405835729)

[4.1.2 Testes de Compressão 33](#_Toc405835730)

[4.1.3 Testes de Iluminação 36](#_Toc405835731)

[5 APRESENTAÇÃO DO SOFTWARE 39](#_Toc405835732)

[5.1 INSTALAÇÃO 50](#_Toc405835733)

[6 CONSIDERAÇÕES FINAIS 51](#_Toc405835734)

[REFERÊNCIAS 52](#_Toc405835735)

[APÊNDICE A - DIAGRAMA DE CASOS DE USO 56](#_Toc405835736)

[APÊNDICE B - ESPECIFICAÇÃO DOS CASOS DE USO 57](#_Toc405835737)

[APÊNDICE C - DIAGRAMA ENTIDADE RELACIONAMENTO 64](#_Toc405835738)

[APÊNDICE D- DIAGRAMA DE CLASSES 65](#_Toc405835739)

[APÊNDICE F - DIAGRAMA DE COMPONENTES 71](#_Toc405835740)

# 1 INTRODUÇÃO

O Programa de Pós-graduação em Bioinformática de Univercidade Federal do Paraná foi criado em 2009 com intenção de atender a grande demanda por profissionais capazes de convergir entre os estudos de biologia molecular e computação, aprofundando-se cada vez mais no campo interdisciplinar chamado Bioinformática.

A Bioinformática é uma ciência que atua na manipulação e extração de conhecimento de dados biológicos. O crescimento da área se deve à necessidade de manipular quantidades enormes de dados genéticos e bioquímicos, utilizando recursos computacionais para catalogá-las, organizá-las e estruturá-las (RASHIDI, 2000).

O PPG-BIOINFO necessita cada vez mais de ferramentas que facilitem o trabalho de pesquisa e armazenamento das informações trabalhadas, e, visando criar um banco de dados para armazenamento das informações genéticas de fungos patológicos, esperamos contribuir de maneira efetiva para que o trabalho de todo o grupo de pesquisa seja facilitado.

Nossa plataforma pretende trazer para o grupo, além de um local para o novo local para armazenamento de informações, uma nova forma de organização de trabalho, possibilitando assim um maior aproveitamento do tempo de seus membros.

# 1.1 OBJETIVOS

Desenvolver uma plataforma web capaz de guardar arquivos genéticos, organizando-os de acordo com o organismo estudado e o projeto em que se encontra. Além disso, fazer com que essa plataforma tenha um sistema de gerenciamento hierarquico de usuários, possibilitando seguir os moldes de organização do PPG-BIOINFO.

1.2 OBJETIVOS ESPECÍFICOS

Este trabalho tem os seguintes objetivos específicos:

* + Possibilitar ao sistema fazer upload de arquivos GFF e GBK
  + Possibilitar ao sistema fazer a conversão de arquivos GFF para GBK e vice-e-versa
  + Possibilitar ao visitante o acesso a dados públicos
  + Possibilitar ao visitante submeter um projeto e cadastro como coordenador a um administrador
  + Possibilitar ao visistante solicitar acesso de pesquisador a um projeto a um coordenador
  + Possibilitar ao administrador do sistema aceitar ou rejeitar solicitações públicas para inclusão de projetos e de acesso como coordenador
  + Possibilitar ao administrador redesignar um projeto a outro coordenador
  + Possibilitar ao administrador editar e excluir coordenadores
  + Possibilitar ao coordenador cadastrar, editar e excluir um projeto
  + Possibilitar ao coordenador editar um equipe
  + Possibilitar ao coordenador aceitar ou rejeitar solicitação de pesquisadores ou convidados para um projeto
  + Possibilitar ao coordenador editar um pesquisador
  + Possibilitar ao coordenador fazer upload de arquivos genéticos de um organismo
  + Possibilitar ao pesquisador acesso aos arquivos em que faz parte do projeto
  + Possibilitar ao pesquisador solicitar acesso de convidado a um projeto a um coordenador

# 2 FERRAMENTAS UTILIZADAS

O objetivo deste capítulo é apresentar os recursos que foram utilizados para o desenvolvimento do projeto.

2.1 BALSAMIQ MOCKUPS

Para a criação da estrutura do wireframe das telas, foi utilizado o software Balsamiq Mockups. Com ele, conseguiu-se uma forma ágil e realista o desenvolvimento do layout modelo, tudo isso devido às facilidades trazidas pela ferramenta (BALSAMIQ, 2016).

2.2 ASTAH PROFESSIONAL E UML

A UML (*Unified Modeling Language*) é uma linguagem visual utilizada para modelar software baseados no paradigma de orientação a objetos. É uma linguagem de modelagem de propósito geral que pode ser aplicada a todos os domínios de aplicação. Essa linguagem tornou-se, nos últimos anos, a linguagem-padrão de modelagem adotada internacionalmente pela indústria de engenharia de software (GUEDES, 2011).

Neste trabalho utilizamos o diagrama de caso de uso, o diagramas de classe e o diagrama de telas. O diagrama de caso de uso permite organizar os requisitos do sistema dando uma visão geral. O diagrama de classes representa a estrutura do sistema orientado a objetos e o diagrama de telas apresenta o fluxo das atividades do aplicativo.

Para a criação desses diagramas utilizamos o Astah, que é uma ferramenta para criação de diagramas UML, a mais de 10 anos disponível no mercado (primeiramente com o nome JUDE). Com ele é possível criar todos os diagramas de uma forma simples, intuitiva e padronizada.

2.4 ORACLE VM VIRTUALBOX

A Virtual Machine da Oracle, VirtualBox, é um software de virtualização. Com ela é possível rodar os mais diversos sistemas operacionais sem a necessidade de modificação, possibilitando assim o desenvolvimento em um sistema mais parecido com o do PPG-BIOINFO do que o disponível em nossas máquinas (ORACLE, 2016).

2.5 MYSQL WORKBENCH

O MySQL Workbench é uma ferramenta para design de banco de dados, também de propriedade da Oracle. Além do design do banco de dados, única funcionalidade a qual utilizamos, a ferramenta integra uma plataforma para desenvolviemnto em SQL, administração de bancos, monitoramento de performance e migração entre bancos (MYSQL, 2016).

# 2.6 POSTGRESQL

PostgreSQL é um SGBD que implementa um banco de dados SQL relacional. O PostgreSQL é um software livre de domínio público e multiplataforma. Este SGBD foi escolhido pela equipe pela necessidade do grupo de Pós-graduação em Bioinformática da UFPR em unificar o desenvolvimento de novas ferramentas sobre um único SGBD, ficilitando assim a manutenção e o aprendizado (POSTGRESQL, 2016).

# 3 METODOLOGIA

A função deste capítulo é apresentar as metodologias e ferramentas que foram utilizadas para o desenvolvimento deste aplicativo. Também serão mostradas as responsabilidades dadas a cada membro.

# 3.1 MODELO DE PROCESSO DE ENGENHARIA DE SOFTWARE

Neste projeto foi utilizado o modelo RUP (*Rational Unified Process*)  que fornece diretrizes para definir as tarefas e atribuir responsabilidades aos membros do projeto. O RUP possui quatro fases: iniciação, elaboração, construção e transição. Na iniciação define-se o escopo do projeto. Na elaboração é obtida uma visão abrangente do sistema, através da construção de protótipos e também é definida a arquitetura do sistema. Na construção, o foco está no desenvolvimento do sistema. Por fim, na transição, o produto é transferido ao usuário e o projeto é avaliado e pode ser concluído. Em todas as fases há o gerenciamento dos requisitos e dos recursos do projeto. Por isso o RUP é baseado no desenvolvimento iterativo, que por sua vez é mais flexível quanto às mudanças de escopo durante o desenvolvimento do projeto (IBM, 2014).

# 3.2 PLANO DE RISCOS

No plano de riscos foram definidos os impactos e a probabilidade de determinados eventos acontecerem, bem como as possíveis ações que poderiam ser tomadas. O plano de riscos do projeto pode ser observado na TABELA 1.

TABELA 1 – PLANO DE RISCOS

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nº** | **Risco** | **Consequência** | **Ação** | **Probabilidade** | **Impacto** | **Classificação** |
| 1 | Falta de tempo para o desenvolvimento do aplicativo. | Impossibilidade de entregar ou atraso na entrega. | Divisão de tarefas e monitoramento do andamento. | Alto | Alto | 7 |
| 2 | Mudança de requisitos do aplicativo. | Necessidade de mudar o direcionamento do projeto. | Mudanças na estrutura do projeto. | Alto | Alto | 7 |
| 3 | Conhecimento limitado em bioinformática. | Dificuldade no desenvolvimento do projeto. | Estudo e pesquisa na internet, livros e em outros materias da PPG-BIOINFO. | Alto | Moderado | 6 |
| 4 | Perda de dados. | Impossibilidade de continuar o projeto. | Manter o repositório do GIT atualizado | Moderado | Alto | 6 |
| 5 | Desistência de um membro da equipe. | Não haveria a possibilidade de entrega do trabalho, devido a resolução do curso. | Conseguir outro membro para o grupo. | Moderado | Alto | 6 |
| 6 | Infraestrutura insuficiente. | Falta de equipamentos para desenvolvimento e testes. | Utilizar emuladores para as necessidades específicas. | Moderado | Moderado | 4 |

# 3.3 MATERIAIS

O objetivo deste capítulo é descrever as ferramentas utilizadas para o desenvolvimento do projeto.

# 3.3.1 Ambiente de Desenvolvimento

Para o desenvolvimento do aplicativo, foram utilizados dois computadores:

* Notebook CyberPower MS16-GC, processador Intel Core i7-4810MQ 2.80GHz, memória RAM 16GB, placa de vídeo Nvidia GTX 860M 4GB, sistema operacional Windows 10 Home;
  + Oracle VM Virtual Box, Ubuntu 16.04
* Notebook Dell Latitude 3450, processador Intel Core i5 5200U 2.20GHz, memória RAM 8GB, placa de vídeo Intel HD Graphics 5500, sistema operacional Windows 8.1 Pro;
  + Oracle VM Virtual Box, Ubuntu 16.04

# 3.3.2 Linguagem de Programação

A linguagem de programação escolhida pelos usuários foi Ruby. É uma linguagem relativamente nova, criada em 1995 por Yuri Matsumoto, podende ser base para diversos paradígmas de programação, o qual escolhemos trabalhar com orientação a objetos, podendo assim aplicar boa parte do conhecimento adquirido enquanto alunos de Análise e Desenvolvimento de Sistemas (RUBY, 2016).

A escolha dessa linguagem se deu pela necessidade do grupo de Pós-graduação em Bioinformática da UFPR em unificar o desenvolvimento de novas ferramentas sobre uma única linguagem, ficilitando assim a manutenção e o aprendizado.

3.3.3 Framework de Desenvolvimento

Para um melhor uso da linguagem Ruby, utilizamos o framework Ruby on Rails durante o desenvolvimento do projeto.

Este framework facilita o desenvolvimento, implantação e manutenção, construindo uma aplicação web dentro da arquitetura MVC (RUBY ON RAILS, 2016).

# 3.3.4 Banco de Dados

Para implementar o banco de dados foi escolhido o PostgreSQL, pois é uma ferramenta de fácil uso, com bastante documentação disponível e bem conceiturada no meio acadêmico e comercial, sendo já utilizado no PPG-BIOINFO em outros projetos.

# 3.4.5 Ferramenta para Modelagem UML

A ferramenta utilizada para criação dos diagramas UML foi o Astah Professional.

# 3.4.6 Ferramenta para Criação do Diagrama Entidade Relacionamento

A ferramenta utilizada para criação do diagrama entidade relacionamento foi MySQL Workbench, software gratuíto para uso não comercial e desenvolvido pela Oracle. Está disponível somente para Windows.

# 3.5 FERRAMENTA PARA CONTROLE DE VERSÃO

O git é um sistema de controle de versão de softwares desenvolvidos por um ou mais desenvolvedores. Ele também é um sistema de gerenciamento de código fonte, ou seja, todos os membros da equipe podem enviar alterações. Todos os arquivos e todo histórico é armazenado num repositório (GIT, 2016).

Neste projeto foi usado o GitHub, que além de gerenciar as versões e as alterações no código, possui um repositório online, facilitando as alterações para os membros da equipe.

# 3.6 RUBYGEMS

Gems são um bibliotecas usadas para resolver diversos tipos de problema. Os desenvolvedores utilizam gems para não precisarem criar uma solução já existente.

  Através de pesquisas foi decidido pelos membros da equipe domente biblioteca externa seria usada no projeto para o melhor aproveitamento das informações genéticas, o BioRuby para poder converter os dados entre os mais diversos formatos necessários.

**4 DESENVOLVIMENTO DO PROJETO**

A ideia inicial do projeto era criar um aplicativo em Android que se comunicaria com um webservice, que também seria desenvolvido, para realizar consultas do BLAST por dispositivos móveis. Foram feitas diversas mudanças de escopo e na escolha de linguagens e bibliotecas ao decorrer do projeto, que causaram alguns problemas com o cronograma.

Como já dito, seria desenvolvido um webservice para comunicar-se com o BLAST e o dispositivo móvel. Inicialmente feito no Octave, o processamento da imagem e o OCR estavam funcionando de forma instável. Mesmo assim, foi desenvolvido um servidor em Python para ligar os processos do Octave à web.

  Após alguns testes, a equipe concluiu que seria inviável continuar com o Octave e decidiu implementar o processamento da imagem e o OCR em Python, assim o webservice ficaria em apenas uma linguagem. O Tesseract-OCR foi escolhido como biblioteca para o OCR, era de uso fácil e rápido. Para consulta ao BLAST foi usada a biblioteca BioPython, que estava funcionando sem problemas, mas o processamento da imagem ainda estava instável e havia a necessidade de um servidor.

A equipe decidiu testar todos esses processos no próprio dispositivo móvel para verificar se não haveria limitações de hardware. Os testes com a biblioteca Tess-two foram satisfatórios, pois obteve bons resultados tanto no processamento da imagem quanto no OCR.

Os resultados do BioJava também foram satisfatórios, necessitando apenas da leitura do XML devolvido pelo BLAST que foi facilmente suprida pela linguagem. Assim o projeto migrou-se totalmente para o Java. Posteriormente houve a necessidade de um banco de dados simples, logo a equipe optou pelo SQLite pela praticidade da ferramenta.

A divisão de tarefas foi mudando de acordo com as mudanças de linguagem, graças ao conhecimento da linguagem dos membros da equipe. Apesar das diversas mudanças, a equipe conseguiu se adaptar a elas.

A equipe decidiu tirar os diagramas de sequência e colocar diagramas de atividade, pois são diagramas mais fáceis de serem lidos e ilustram o que ocorre de forma mais simples.

**5 APRESENTAÇÃO DO SOFTWARE**

Neste capítulo será apresentada uma visão geral do software desenvolvido, mostrando suas telas e funcionalidades.

5.1 PERFIL PÚBLICO DE ACESSO

O acesso de todos usuários iniciasse pela tela da FIGURA 01



FIGURA 1 - TELA INICIAL PÚBLICO

Fonte: Os autores (2016).

Ao clicar em “Projects”, temos a tela da FIGURA 02. Apesar dessa tela estar disponível para todos os usuários, os projetos podem possuir restrição de visualização.



FIGURA 2 – TELA PROJETOS

Fonte: Os autores (2016).

E também é possível pedir acesso a um projeto. Caso o usuário esteja logado, é pedido um acesso de convidado (pois ele já é um pesquisador), caso o usuário não esteja logado é pedido um acesso de pesquisador e prossegue para a tela do FIGURA 5.



FIGURA 3 - TELA SOLICITAR ACESSO DE PESQUISADOR

Fonte: Os autores (2016).

É possível baixar os arquivos genéticos de acordo com as restrições de permissão.



FIGURA 4 - TELA ARQUIVOS

Fonte: Os autores (2016).

É preenchido os dados do solicitante e mostrado quais são os dados do proejto solicitado.



FIGURA 5 - TELA CADASTRO PESQUISADOR

Fonte: Os autores (2016).

Ao clicar em “Submit Project” temos a tela da FIGURA 6 para solicitar o acesso como coordenador e depois poder submeter um projeto.



FIGURA 6 - TELA SOLICITAR ACESSO DE COORDENADOR

Fonte: Os autores (2016).

5.2 ACESSO DE ADMINISTRADOR

Depois de logado, é apresentado ao administrador a tela da FIGURA 7, mostrando as possibilidades de administração.



FIGURA 7 - TELA INICIAL ADMINISTRADOR

Fonte: Os autores (2016).

Caso escolha por editar um projeto, o administrador pode mudar o coordenador de um projeto.



FIGURA 8 - TELA EDITAR PROJETOS

Fonte: Os autores (2016).

Na tela da FIGURA 9 o administrador pode aceitar novos coordenadores e novos projetos.



FIGURA 9 - TELA ACEITAR PROJETOS E NOVOS COORDENADORES

Fonte: Os autores (2016).

Caso escolha por editar ou excluir um coordenador, é possível ao administrador efetua essa ação na tela da FIGURA 10.



FIGURA 10 - TELA EDITAR/ EXCLUIR COORDENADORES

Fonte: Os autores (2016).

5.3 ACESSO DE COORDENADOR

Depois de logado, é apresentado ao coordenador a tela da FIGURA 11, mostrando as possibilidades de administração.



FIGURA 11 - TELA

Fonte: Os autores (2016).

Ao clicar em editar um projeto é possível ao coordenador editar as informações de um projeto ou exclui-lo.



FIGURA 12 - TELA EDITAR/EXCLUIR PROJETOS

Fonte: Os autores (2016).

Ao clicar em editar um organismo é possível ao coordenador editar as informações de um organismo ou exclui-lo.



FIGURA 13 - TELA EDITAR/EXCLUIR ORGANISMOS

Fonte: Os autores (2016).

Enquanto edita um organismo o coordenador pode incluir um novo arquivo para aquele organismo, como mostrado na tela da FIGURA 14.



FIGURA 14 - TELA INCLUIR ARQUIVOS DE PROJETO

Fonte: Os autores (2016).

Após submeter um novo projeto (FIGURA 15) o coordenador monta a equipe para aquele projeto, como mostrado na tela da FIGURA 16



FIGURA 15 - TELA SUBMETER UM NOVO PROJETO

Fonte: Os autores (2016).



FIGURA 16 - TELA CADASTRA EQUIPE

Fonte: Os autores (2016).

Ao receber uma solicitação de acesso de pesquisador/convidado a um de seus projetos, o coordenador pode decidir se aceita ou não.



FIGURA 17 - TELA ACEITAR PESQUISADORES/CONVIDADOS

Fonte: Os autores (2016).

O coordenador também pode excluir seus pesquisadores, como visto na tela da FIGURA 18.



FIGURA 18 – TELA EXCLUIR PESQUISADORES

Fonte: Os autores (2016).

O coordenador também consegue editar as equipes de seus projetos



FIGURA 19 - TELA EDITAR EQUIPES

Fonte: Os autores (2016).

5.4 ACESSO DE PESQUISADOR

No momento as atividades possíveis a um pesquisador não varia muita da que a de um usuário pública, a única diferença no momento é a possibilidade de ver mais projetos devido a sua restrição.

# 5.5 INSTALAÇÃO

Para instalar o aplicativo, deve-se entrar no link do aplicativo na Play Store (https://play.google.com/store/apps/details?id=com.dnashot.v0), mostrado na FIGURA X1, toque em “Instalar” e depois em “Aceitar”. Após o download e a instalação do aplicativo, ele já estará pronto para uso. A equipe recomenda o uso deste aplicativo em celulares com processador de 1 GHz ou mais, além de 1GB de memória RAM.



FIGURA xx - TELA

Fonte: Os autores (2016).

**6 CONSIDERAÇÕES FINAIS**

O desenvolvimento do sistema nos possibilitou ter um contato inicial com a area da bioinformática e perceber a importância dessa tanto para o desenvolvimento das mais diversas areas da biologia, apesar do nosso foco em genética, como para novas tecnologias.

O aplicativo consegue cumprir as funções de armazenamento de informações genéticas dos organimos pesquisados assim como auxilar no gerenciamentos de times e projetos.

  As funcionalidade de download e upload de arquivos genéticos consome um alto valor computacional e de rede, por isso o sistema pode vir a enfrentar problemas de lentidão em casa de muitos acessos simultâneos a essas funções.

Durante o desenvolvimento, várias ideias de como o software poderia ser melhorado e ampliado surgiram, porém não foi possível implementá-las devido a complexidade e tempo. Algumas destas ideias para uma futura versão estão listadas abaixo:

* Criar uma ferramenta para busca por genomas ou descrição de genes
* Utilizar um banco de dados non-SQL afim de diminuir o impacto computacional para Download e Upload

**REFERÊNCIAS**

ASTAH. **Astah – Fundamentals**. Disponível em: <http://astah.net/fundamentals>. Último acesso em: 27/06/2016.

BALSAMIQ. **Balsamiq Mockups**. Disponível em: <https://balsamiq.com/products/mockups/>. Último acesso em: 26/06/2016.

BIORUBY. **BioRuby: bioinformatics software for the Ruby programming language**. Disponível em: < http://bioruby.org/>. Último acesso em: 20/06/2016.

BROAD INSTITUTE. **Black Yeast Database.** Disponível em: < http://www.broadinstitute.org/annotation/genome/Black\_Yeasts/MultiHome.html>. Último acesso em: 26/06/2016.

GIT. **About – Git.** Disponível em: <http://git-scm.com/about/>. Último acesso: 20/05/2016.

GUEDES, G. **UML 2: uma abordagem prática**. São Paulo: Novatec, 2011.

IBM. **IBM – Definição de RUP.** Disponível em: <http://www-01.ibm.com/software/rational/rup/>. Último acesso: 21/05/2016.

MYSQL. **MySQL :: MySQL Workbench**. Disponível em: <https://www.mysql.com/products/workbench/>. Último acesso em: 25/06/2016.

ORACLE. **Oracle VM VirtualBox**. Disponível em: <https://www.virtualbox.org/>. Último acesso em: 24/06/2016.

POSTGRESQL. **PostgreSQL: About**. Disponível em: <https://www.postgresql.org/about/>. Último acesso em: 23/06/2016.

RUBY. **About Ruby**. Disponível em: <https://www.ruby-lang.org/en/about/>. Último acesso em: 20/06/2016.

RUBY ON RAILS. **Ruby on Rails**. Disponível em: <http://rubyonrails.org/>. Último acesso em: 20/06/2016.

Astah Professional

Disponível em: < http://astah.net/download>. Último acesso em: 27/06/2016.

Balsamiq Mockups

Disponível em: < https://balsamiq.com/download/>. Último acesso em: 26/06/2016.

GitHub

Disponível em: < https://github.com/join>. Último acesso: 20/05/2016.

MySQL Workbench

Disponível em: < https://www.mysql.com/downloads/>. Último acesso em: 25/06/2016.

Oracle VM VirtualBox

Disponível em: < https://www.virtualbox.org/wiki/Downloads>. Último acesso em: 24/06/2016.

PostgreSQL

Disponível em: < https://www.postgresql.org/download/>. Último acesso em: 23/06/2016.

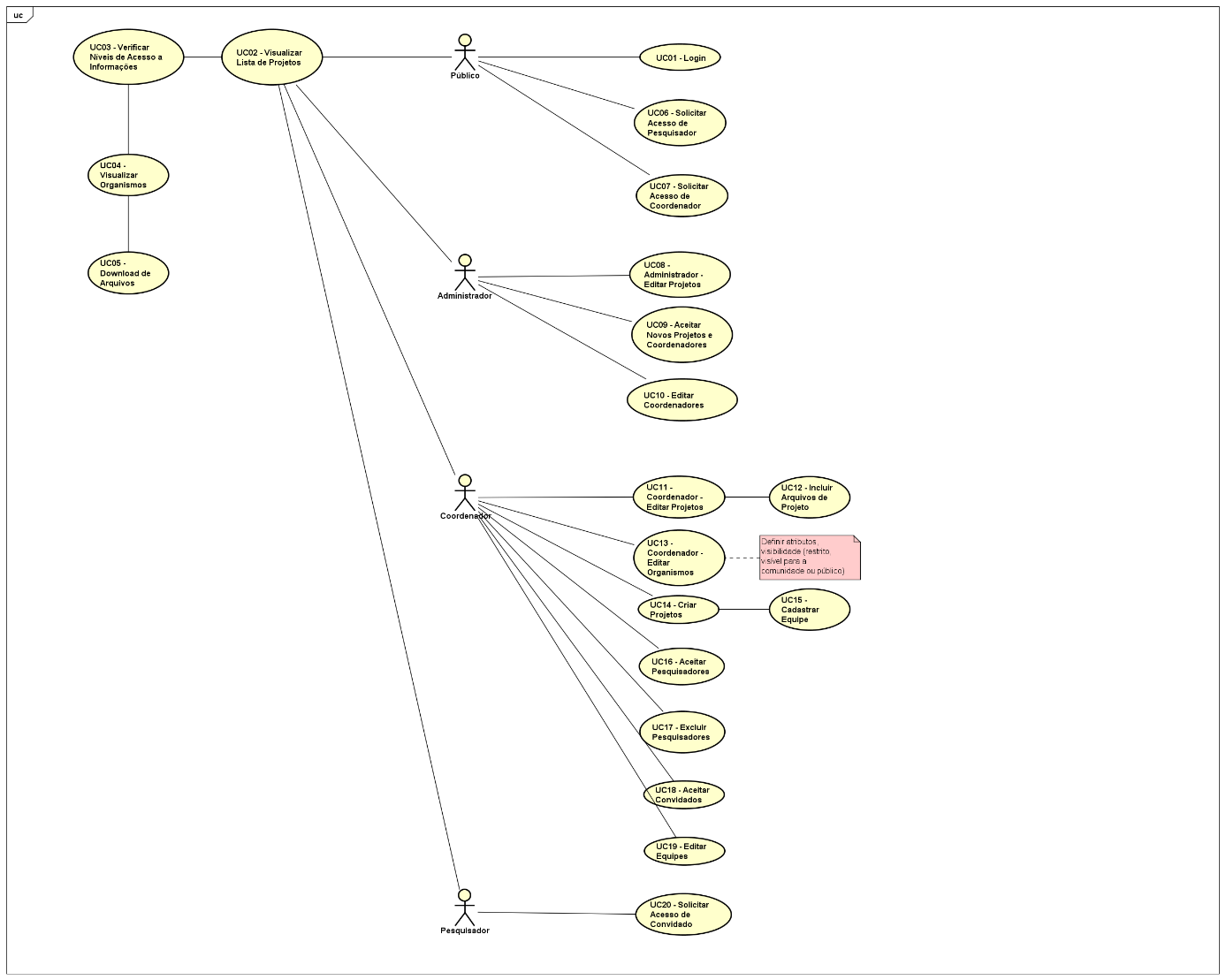
**APÊNDICE A - DIAGRAMA DE CASOS DE USO**

FIGURA xx – DIAGRAMA DE CASOS DE USO

Fonte: Os autores (2016).

**APÊNDICE B - DIAGRAMA ENTIDADE RELACIONAMENTO**

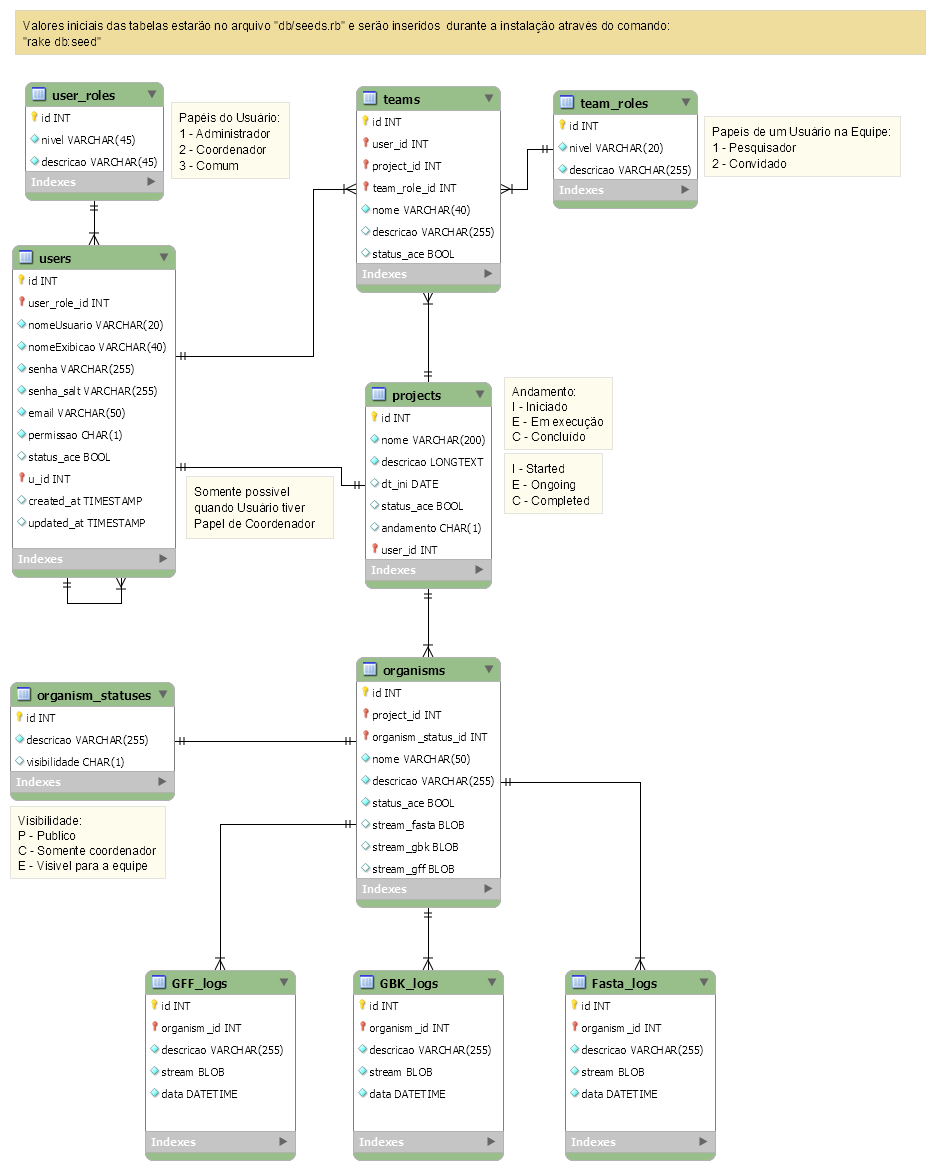


FIGURA xx – DIAGRAMA ENTIDADE RELACIONAMENTO

Fonte: Os autores (2016).

**APÊNDICE C- DIAGRAMA DE TELAS**

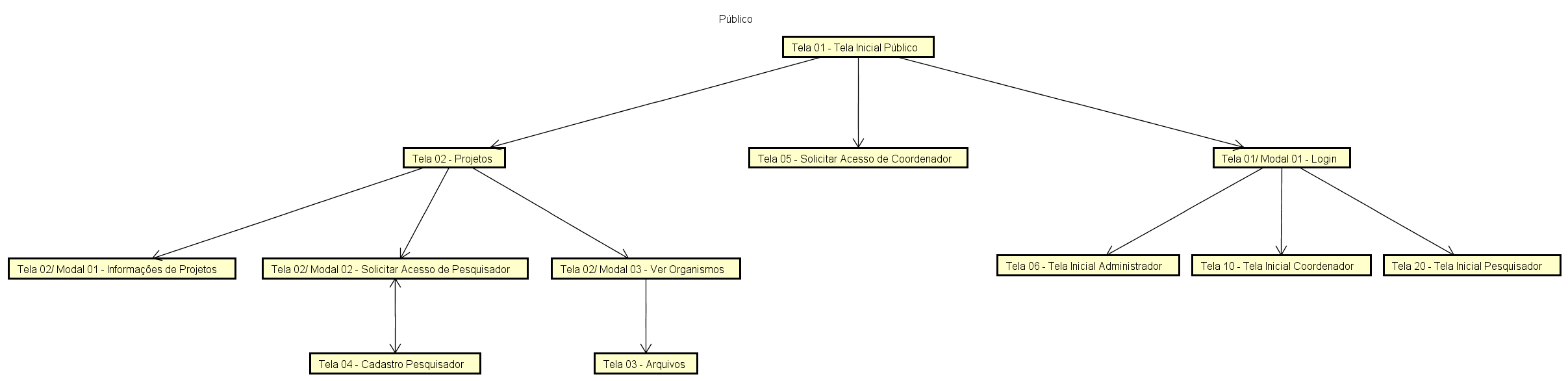


FIGURA xx – DIAGRAMA DE TELAS – ACESSO PÚBLICO

Fonte: Os autores (2016).

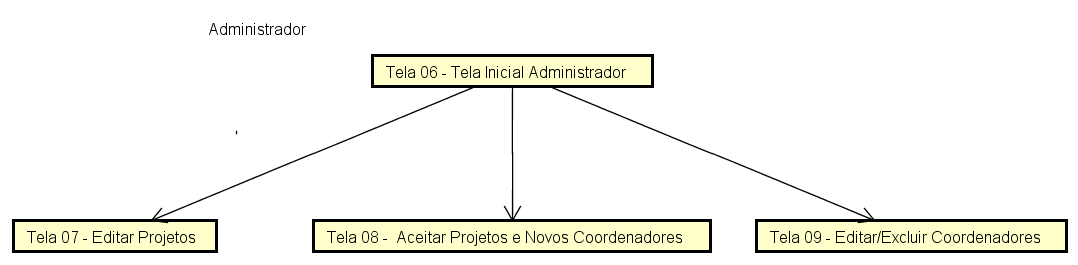


FIGURA xx – DIAGRAMA DE TELAS – ACESSO DE ADMINISTRADOR

Fonte: Os autores (2016).

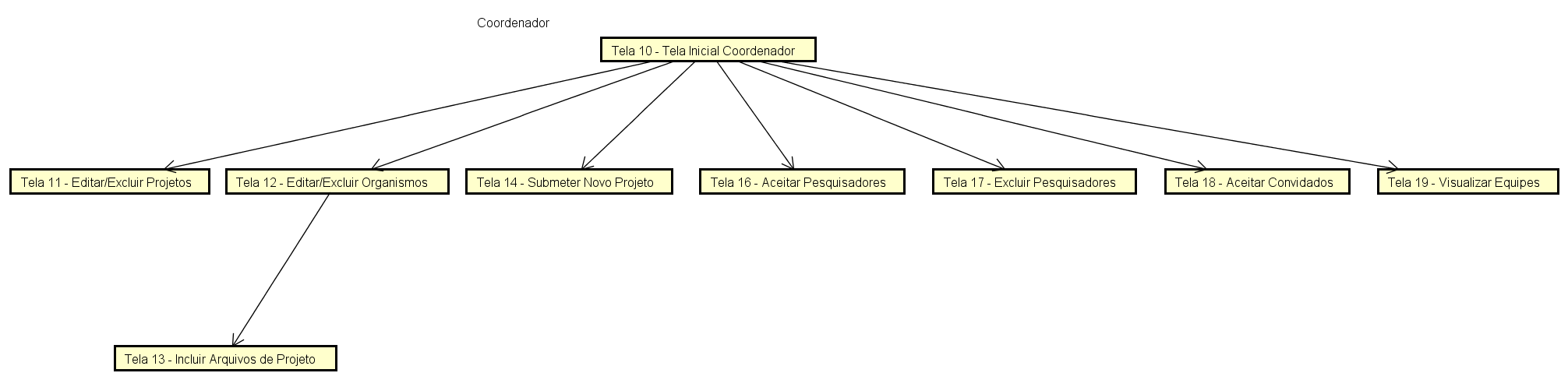


FIGURA xx – DIAGRAMA DE TELAS – ACESSO DE COORDENADOR

Fonte: Os autores (2016).

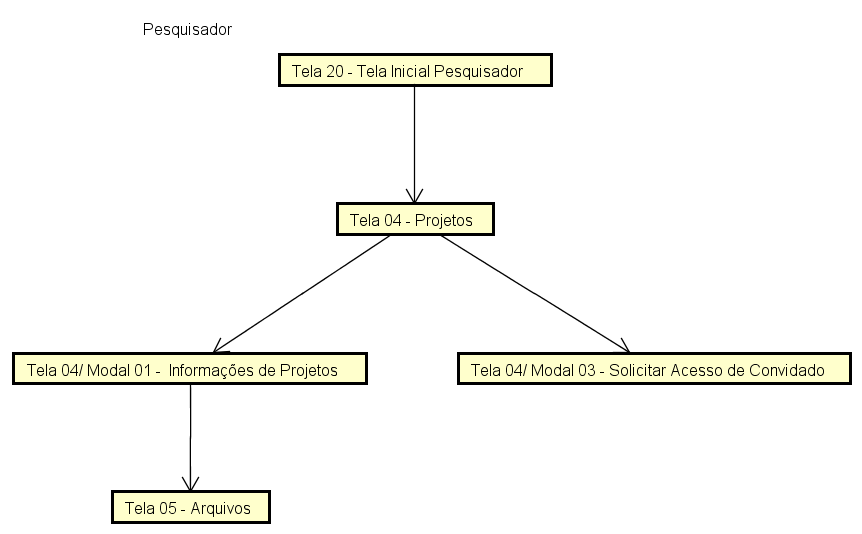


FIGURA xx – DIAGRAMA DE TELAS – ACESSO DE PESQUISADOR

Fonte: Os autores (2016).

**APÊNDICE D- DIAGRAMA DE CLASSES**

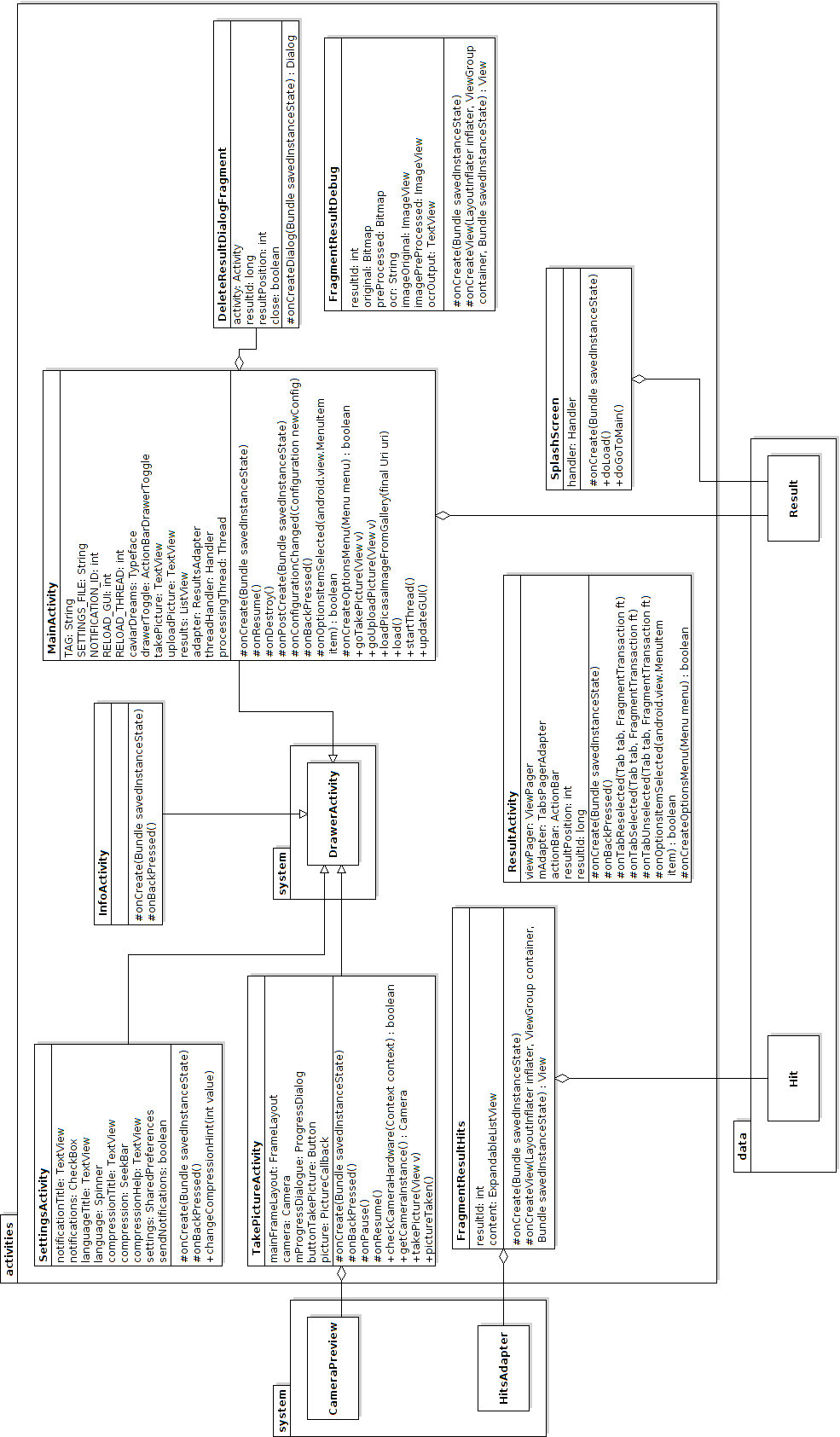
****

FIGURA xx – DIAGRAMA DE CLASSES

Fonte: Os autores (2016).