

Licenciatura em Engenharia Informática

Pomar Digital

Relatório do TFC

Orientador:

Alexandre Pereira

Alunos:

Rúben Querido, nº 20075750

Osmar Seguro, nº 20083842

Resumo

No âmbito da cadeira de Projeto de Final de Curso, do 3º ano da Licenciatura em Engenharia Informática da ULHT, foi formado um grupo de trabalho constituído pelos alunos Osmar Seguro e Rúben Querido.

Com base nos nossos conhecimentos adquiridos e pelos nossos interesses, decidimos realizar uma aplicação móvel para dispositivos com sistema operativo Android.

A aplicação permite fazer o registo de dados no dispositivo e a sincronização com o servidor de Base de Dados, que depois podem ser consultadas tanto num dispositivo móvel como numa aplicação Web.

Abstract

As part of the discipline Final Project Course, 3rd year of Bachelor in Computer Engineering from ULHT, we formed a working group composed by the students Osmar Seguro and Ruben Querido.

Based on our knowledge and our interests we decided to make an application for mobile devices with Android operating system.

The application lets you record data on the device and sync with the Database server, which can then be viewed on a mobile device such as on a web application.

Agradecimentos

Foram algumas as pessoas que participaram e acompanharam a realização deste projeto e às quais estamos profundamente agradecidos:

Ao professor Alexandre Pereira, pela forma prestável com que orientou o nosso projeto, pelos ensinamentos e dúvidas esclarecidas, que ajudaram à realização do mesmo.

Ao professor Mauro Bragança, pelas ideias inovadoras e pela atenção e disponibilidade demonstradas no decorrer da realização deste trabalho.

A todos os professores do curso de Engenharia Informática da Universidade Lusófona de Humanidades e Tecnologias, que contribuíram para a nossa formação, tendo um papel fundamental em todo o nosso percurso até à realização deste projeto.

A todos os nossos colegas, que nos acompanharam e que, de alguma forma, foram importantes para a realização do nosso projeto.

À nossa família, pela compreensão e paciência que tiveram ao longo destes anos, e em especial nos últimos meses, com que viram crescer muito deste trabalho.

A todos o nosso Muito Obrigado!

Índice

Introdução	6
Capítulo I -Descrição da aplicação.....	7
Tecnologias usadas.....	8
CAPÍTULO II – Trabalho realizado	10
Análise de requisitos	10
Funcionalidades da aplicação.....	11
Base de Dados da aplicação	11
O sistema de Login.....	12
O processo de Sincronização.....	15
Código usado:.....	16
Aplicação Web	19
JSCharts.....	22
Conclusão	24
Bibliografia	25
Anexos.....	26
Configuração da biblioteca ZXing para utilização do QRCODE	27
Source Code - Aplicação Web	33
Source Code - Aplicação Java para Android.....	60

Índice de figuras

Ilustração 1 - Esquema App.....	7
Ilustração 2 - Base de Dados da aplicação	12
Ilustração 3 - Esquema do Login	14
Ilustração 4 - Esquema da Sincronização	18
Ilustração 5 - HomePage da Aplicação Web	19
Ilustração 6 - Edição de Informação.....	19
Ilustração 7- Gráfico de Limpezas	20
Ilustração 8 – EsquemaWebPage.....	21
Ilustração 9 - ZXing - Novo Projecto	28
Ilustração 10 - ZXing - Target.....	29
Ilustração 11 - ZXing Properties	30
Ilustração 12 - ZXing - Java Build Path.....	30
Ilustração 13 - ZXing Resume	31

Introdução

A Informática tem evoluído drasticamente com o passar dos tempos e cada vez mais é um recurso para as mais diversas áreas de negócio. Tem representado um papel importante para a automatização e processamento de informação de tarefas cada vez mais complexas. O uso de dispositivos móveis tem sido cada vez mais útil e explorado pelas empresas.

Na sequência desta evolução decidimos apostar na investigação e exploração da tecnologia Android para o desenvolvimento de uma aplicação com o objetivo de registar informação relativa a árvores de fruto.

No **Capítulo I**, pretendemos descrever, de forma geral, o projeto que desenvolvemos e apresentar as tecnologias e ferramentas a que recorremos para a realização do mesmo.

O **Capítulo II** representa a parte principal do relatório, onde apresentamos, com detalhe, toda a arquitetura, a forma como esta foi pensada, assim como todo o desenvolvimento da aplicação. Quando necessário, e de modo a tornar mais clara a informação, recorreremos também a exemplos, esquemas gráficos e figuras.

Na **Conclusão**, descrevemos os objetivos alcançados relativamente à aplicação em questão, a sua utilidade e possíveis alterações futuras com vista a incluir outras funcionalidades.

Finalmente, em **Anexo**, constam todas as informações detalhadas relativamente à implementação do projeto.

Capítulo I -Descrição da aplicação

É nosso objetivo passa pela criação de uma aplicação para telemóveis com sistema operativo android (App) que permite guardar informação de árvores de fruto. As linguagens utilizadas serão JAVA para android, HTML, PHP, JAVASCRIPT e SQL. A plataforma de desenvolvimento para JAVA é o ECLIPSE.

As árvores irão ser identificadas através da implementação de um identificador gerado em QRCODE, que é lido pela câmara do equipamento. Essa identificação será usada para a BD da árvore.

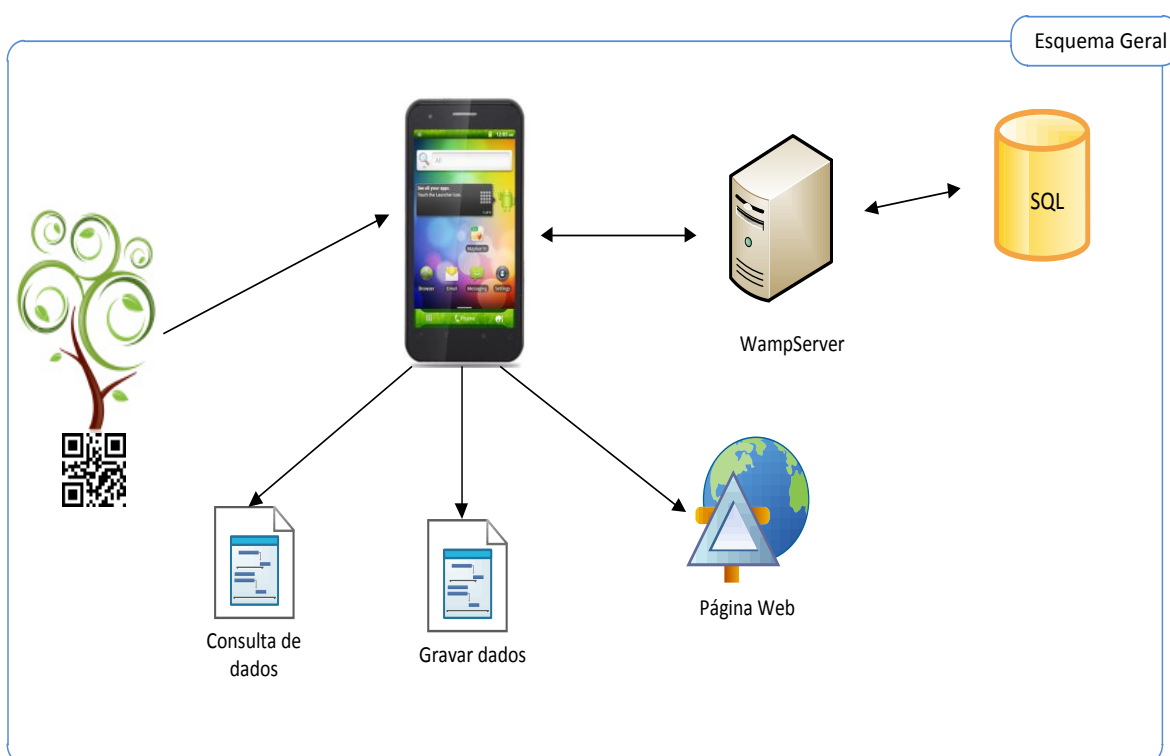


Ilustração 1 - Esquema App

Tecnologias usadas

QRCode e ZXing

QRcode, ou código QR, é um código de barras bidimensional que pode ser facilmente lido pela maioria dos dispositivos móveis equipados com câmara.

Esse código pode ser convertido num pedaço de texto (interativo), num endereço URL, num número de telefone, numa localização georreferenciada, num e-mail, num contacto ou num SMS, entre outros exemplos. No nosso caso, serviu como identificador das árvores que vão ser registadas.

No projeto, vamos usar a livreria ZXing. ZXing (pronuncia-se "zebra crossing") é uma biblioteca, open-source, de processamento de imagem implementada em Java, compatível para outras línguas. Suporta o multi-formato de código de barras 1D/2D. O nosso objetivo é usar a câmara do dispositivo móvel para ler e decodificar códigos de barras no aparelho, sem ser necessário comunicar com um servidor.

SQL e SQLite

SQL é uma linguagem de Base de Dados relacional. É a linguagem padrão nesta área da programação, devido à sua simplicidade e facilidade de uso. Através do uso de uma linguagem simples e interativa torna-a fácil de uso e de aprendizagem, sendo a linguagem por onde se começa na área de Base de Dados.

No nosso projeto, foi utilizada para construir toda a parte de armazenamento de dados do lado do servidor.

Alguns comandos SQL:

DML – Data Manipulation Language (Manipular Dados)

`SELECT * FROM tabela` - Selecionar todos os dados de uma dada tabela.

`INSERT into tabela (dado1, dado2, dadon)` - Insere dados numa tabela.

`UPDATE tabela SET coluna1 = 'dados', coluna2 = 'dados' WHERE 'condição'` - Update de dados

`DELETE FROM table WHERE 'condição'` - Apagar dados da tabela

DDL – Data Definition Language (Definir Dados)

`CREAT tabela` - criar nova tabela

`ALTER tabela` - alterar tabela

`DROP tabela` - apagar tabela

DQL – Data Query Language (Consultar Dados)

É a parte mais utilizada no SQL, permitindo executar uma QUERY (consulta de dados), com o comando SELECT e, utilizando as seguintes cláusulas, podemos especificar a consulta que queremos fazer:

Cláusulas

FROM - Especificar uma tabela

WHERE - Especificar uma condição do registo que queremos obter

GROUP BY - Separar em grupos específicos

HAVING - Expressar a condição que cada grupo deve satisfazer

ORDER BY - Ordenar registos numa forma específica

DISTINCT - Seleciona dados sem repetição

SQLite é uma biblioteca da linguagem C que faz a implementação de Base de Dados SQL. É usada para construir e guardar dados diretamente no dispositivo onde é criada.

No nosso caso, é o que serve de suporte ao armazenamento dos dados nos dispositivos móveis, sendo depois sincronizada com SQL do servidor.

JSON (Java Script Object Notation) é um subconjunto de JavaScript, mas a sua utilização não requer o uso de JavaScript exclusivo. É uma forma de criar estrutura de dados de forma simples e rápida. No nosso projeto, foi utilizada para a criação de objetos JSON para sincronização de dados entre as duas Bases de Dados.

Exemplo

```
("INSERT INTO arvores  
(numArvore,Data_Plantacao,Data_Implementacao,Tipo_Arvore,Produtor_idProdutor)  
VALUES ('".$jsjsons->{'IDArvore'}."', '".$jsjsons->{'Data_Plantacao'}."', '".$jsjsons->{'Data_Implementacao'}."', '".$jsjsons->{'Tipo_Arvore'}."', '".$jsjsons->{'idProdutor'}."')");
```

Análise de requisitos

- Definir dados de utilizador para login na aplicação através de um pedido prévio de registo aos administradores da aplicação. Esses dados poderão ser alterados pelo utilizador na página web.
- Efectuado o login na aplicação e se tiver uma ligação á internet ativa é efectuado uma sincronização bidimensional dos dados com o servidor web.
- O registo das árvores é feito através da leitura do código (QRCode) presente na árvore e gravado na base de dados interna com a informação referente á mesma. (Data de Plantação, Data de implementação, Tipo de árvore).
- Para cada árvore existem n intervenções. Essas intervenções são realizadas através da leitura do código (QRCode) presente nas árvores e gravadas na base de dados interna com a informação referente á mesma. (Data de intervenção, Motivo (Colheita, Limpeza, Tratamento, Outro), Observação)
- O utilizador pode consultar os dados gravados referentes às árvore e às intervenções realizadas nas mesmas. As consultas são realizadas através de filtros (Todas, Intervenção, Motivo).
- O utilizador pode sincronizar a base de dados no dispositivo, caso exista uma ligação de dados ativa, com a base de dados externa (servidor).
- O utilizador pode aceder á página Web através do seu dispositivo.

Funcionalidades da aplicação

A aplicação inicia-se com o menu principal, onde é possível aceder diretamente à aplicação Web ou, então, fazer login com os dados do produtor para poder fazer a leitura de árvores e a sincronização dos dados que possam já estar armazenados no dispositivo. Ao seleccionar a sincronização, tudo o que estiver no telemóvel é enviado para o servidor e vice-versa (caso tenha internet). Quando é seleccionada a opção para ler árvores, é ativado o leitor de QRcode e passa a ser possível fazer a leitura do identificador da árvore. Quando a leitura é efetuada com sucesso, são enviados para a ficha de submissão os dados relativos ao ID da árvore e ao ID do produtor. Depois de preenchida toda a informação, esta fica guardada no dispositivo para posterior sincronização com o servidor Web. É possível fazer também diversos tipos de consulta de dados já armazenados (por árvore, por intervenção e por tipos de intervenção).

Base de Dados da aplicação

A nossa Base de Dados assenta na criação de três tabelas, referentes às Árvores, Produtor e Intervenções. Na tabela *Produtor*, temos todos os dados relativos aos utilizadores da aplicação, que podem ser alterados no interface Web. Esta tabela tem uma ligação do tipo 1:N com a tabela *Árvores*, isto porque um produtor tem várias árvores no seu registo. A tabela *Árvores* contém toda a informação relativa a cada árvore. Esta tabela tem uma ligação à tabela *Intervenções*, onde são armazenados todos os registos de operações efetuadas em cada árvore. Esta tabela tem uma ligação do tipo 1:N, pois podem ser feitas várias intervenções numa árvore ao longo do seu ciclo vida.

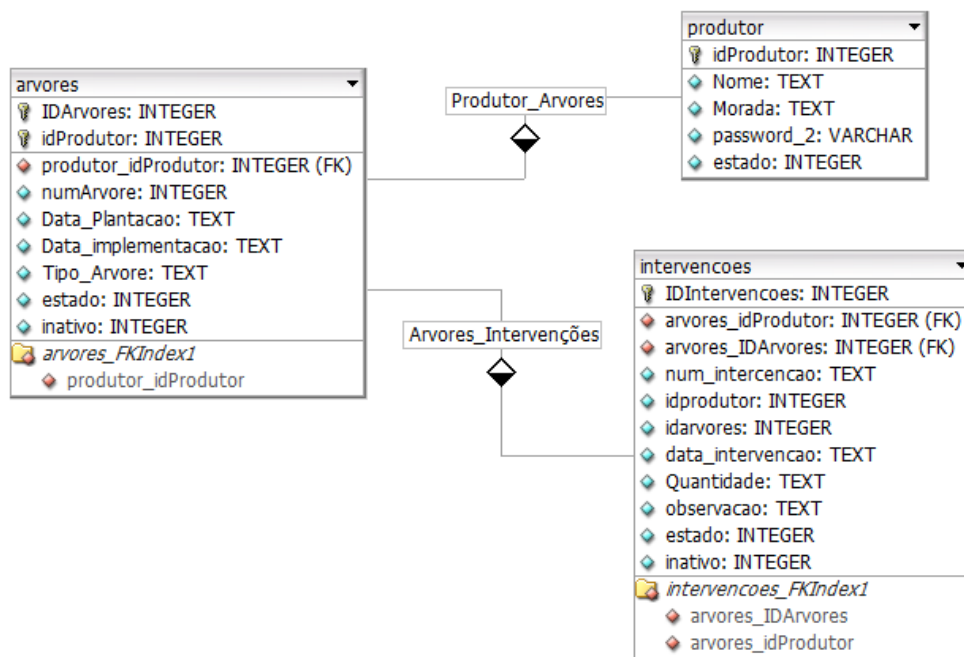


Ilustração 2 - Base de Dados da aplicação

O sistema de Login

O utilizador tem que efetuar o Login na aplicação para usufruir das suas funcionalidades. Para isso, terá de ser realizado um registo prévio na aplicação Web, através do contacto com os administradores da plataforma.

Ao premir o botão “Login”, a aplicação vai pedir as credenciais ao utilizador e quando premir o botão “OK” vai verificar se esse utilizador já tem o registo efetuado, ou seja, se for o primeiro login, o utilizador não tem os seus dados registados na base de dados interna do dispositivo (SQLite), então é necessário conexão à internet para efetuar a sincronização dos dados presentes na base de dados externa (WampServer). Se existirem dados na Base de Dados externa, então faz a sincronização e grava na Base de Dados interna (SQLite).

Depois de um Login válido, o utilizador entra no menu onde são mostrados os botões das funcionalidades implementadas. Caso contrário, o utilizador não tem acesso a qualquer funcionalidade da aplicação e é direcionado para a HomePage da aplicação.

```

public void VerificaLogin()
{

    btok.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            try {
                cursor = BD.query("users", new String []
{"IDUser", "Password"},
                "IDUser like '"+etuser.getText().toString()+"' and Password like
                '"+etpass.getText().toString()+"'", //faz busca pelas credenciais,
                null, //selectionArgs,
                null, //groupBy,
                null, //having,
                null,
                null); // Limite de registros retornados

                int numeroRegistros = cursor.getCount();
                if (numeroRegistros != 0) // Se existirem registros na bd
interna
                {
                    id = etuser.getText().toString();
                    login=true;
                    main();// Mostra a "HomePage" com as opcoes todas

                }
                else
                {
                    LoginServer();// Função que executa a tarefa de
sincronização com o servidor
                }
            }
            catch (Exception erro) {
                msg("Erro Banco", "Erro buscar dados no banco:
                "+erro.getMessage());
            } } } );

```

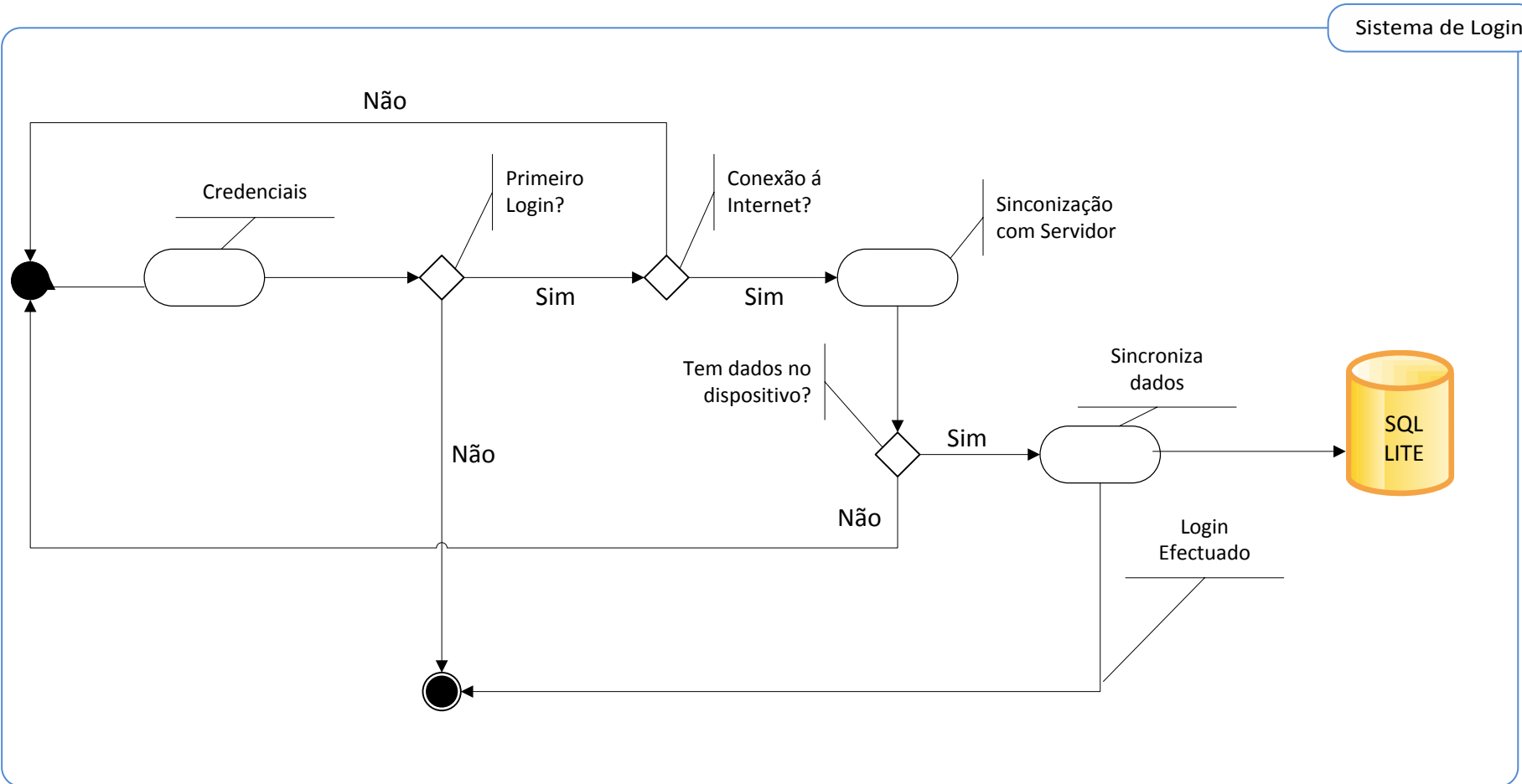


Ilustração 3 - Esquema do Login

O processo de Sincronização

Ao entrar na aplicação pela primeira vez, é necessário fazer a sincronização com os dados armazenados na Base de Dados do servidor, seja já um utilizador com dados inseridos ou um novo utilizador.

Para o caso de novos utilizadores, esta sincronização é necessária para poder fazer o Login na aplicação e começar a fazer o registo de dados no dispositivo.

Nos restantes, é garantido que todos os dados recolhidos pelo dispositivo são sincronizados com a Base de Dados no servidor, para que não haja perda de dados e para que toda a informação recolhida fique disponível para consulta na Aplicação Web.

A aplicação tem a funcionalidade de sincronização dos dados, para que seja possível ao utilizador gravar todos os dados recolhidos. Para isso tem de ter conexão à internet e ativar essa funcionalidade, sendo feita automaticamente a sincronização das duas Base de Dados (SQLite com SQL do servidor).

O processo começa com uma requisição ao servidor, através de um PHP que executa a Query de ligação e de pedido dos dados; este recebe-os e encapsula-os num objecto JSON.

A sincronização de dados pode ser uma tarefa bastante demorada, dependendo do número de registos que se queiram sincronizar. Se tentarmos sincronizar do modo convencional, o mais provável é que a aplicação “trave”. Para fazermos uma sincronização “limpa” recorremos à classe `AsyncTask` (tarefa assíncrona). A `AsyncTask` possibilita um uso fácil e apropriado do UI thread. Esta classe permite a execução de operações em background. A `AsyncTask` executa operações de curta duração que de outra forma poderiam demorar demasiado tempo. É uma classe que serve para dar suporte aos eventos de Thread e Handler e não constitui uma plataforma de processamento de threads. Uma tarefa assíncrona é definida por um cálculo que funciona numa thread em “background” e cujo resultado é publicado na UI thread. Uma tarefa assíncrona é definida por três tipos genéricos, chamados **Params**, **Progress**, e **Result**, e quatro etapas, chamadas **OnPreExecute**, **doInBackground**, **onProgressUpdate** e **OnPostExecute**.

Código usado:

```
private class LoginServerTask extends AsyncTask<String, Void, Void> {
    ProgressDialog dialog;
    boolean erro=false,erroServer=false,erro_Login=false;
    @Override
    protected void onPreExecute() //mostra o dialog enquanto estiver
a executar a tarefa
    {
        dialog = new ProgressDialog (AppPomar.this);
        this.dialog.setMessage("A verificar dados...");
        this.dialog.show();
        this.dialog.setCancelable(false);
    }

    // acaba automaticamente
    @Override
    protected Void doInBackground(String... args)//estes args são os
urls das paginas php
    {
        ArrayList<BasicNameValuePair> postParameters = new
ArrayList<BasicNameValuePair>();
        postParameters.add(new BasicNameValuePair("user",
etuser.getText().toString()));
        postParameters.add(new BasicNameValuePair("pass",
etpass.getText().toString()));
        String response = null; String resposta =null;
        try {
            if(serverIsOnline())//funcao que verifica se o servidor
está online
            {
                response = CustomHttpClient.executeHttpPost(args[0],
postParameters);
                resposta = response.toString();
                resposta = resposta.replaceAll("\\s+", "");
                Log.i("VERIFICA RESPOSTA",response.toString());
                if(resposta.equals("1")) //faz a verificação se existem
dados no servidor atraves da resposta vinda da página php
                {
                    InsereUser(); //insere na bd interna
                    id = etuser.getText().toString();
                    pass=etpass.getText().toString();
                    recebeDados(args[1]); //recebe os dados referentes
às árvores
                    recebeIntervencoes(args[2]); //recebe os dados
referentes às intervenções nas árvores
                    erro_Login=false;
                }
            }
            else
            {
                if(erroLogin<=3)
                {
                    erroLogin++;
                    Log.i("ERROLOGIN", ""+erroLogin);
                    erro_Login=true;
                }
            }
            else
            {
                erro=true;
            }
        }
    }
}
```



```

    }
    }
    else{
        erroServer=true;
    }
    }

    catch (Exception e) {

    }

    return null;
}

@Override
protected void onPostExecute(Void unused) //esta função é
executada depois da função principal.
{
    Log.i("ERRO_LOGIN", ""+erroLogin);
    if(erroServer==true)
    {
        msg("Erro!", "Servidor encontra-se em modo offline. Tente
mais tarde.");
        dialog.dismiss();
    }
    else
    {
        if(erro_Login==false)
        {
            dialog.dismiss();
            login=true;
            main();
            msg("Concluído", "Login Correto. Dados sincronizados!");
        }
        else if((erro_Login==true) && (erroLogin<=3))
        {
            msg("Erro", "Login incorrecto!");
            dialog.dismiss();
        }
        else{ //if((erro==true)&&(erroLogin==3))
            msg("Erro", "Efectue o registo na app web, clicando
no botão 'AppWeb'");
            dialog.dismiss();
        }
    }
}

}

public void LoginServer() {
    if(isOnline())//Funcao que verifica se o dispositivo tem
internet activa
    {
        LoginServerTask task = new LoginServerTask(); //Cria a
tarefa
        task.execute(new String[] {
"http://192.168.2.101/login.php", "http://192.168.2.101/getArvores.php"
, "http://192.168.2.101/getIntervencoes.php" });
    }
    else
    {
        msg("Aviso!", "Necessita de conexão á internet para
verificar se existe no servidor. Caso contrário faça o registo da
árvore na aplicação.");
    }
}
}

```

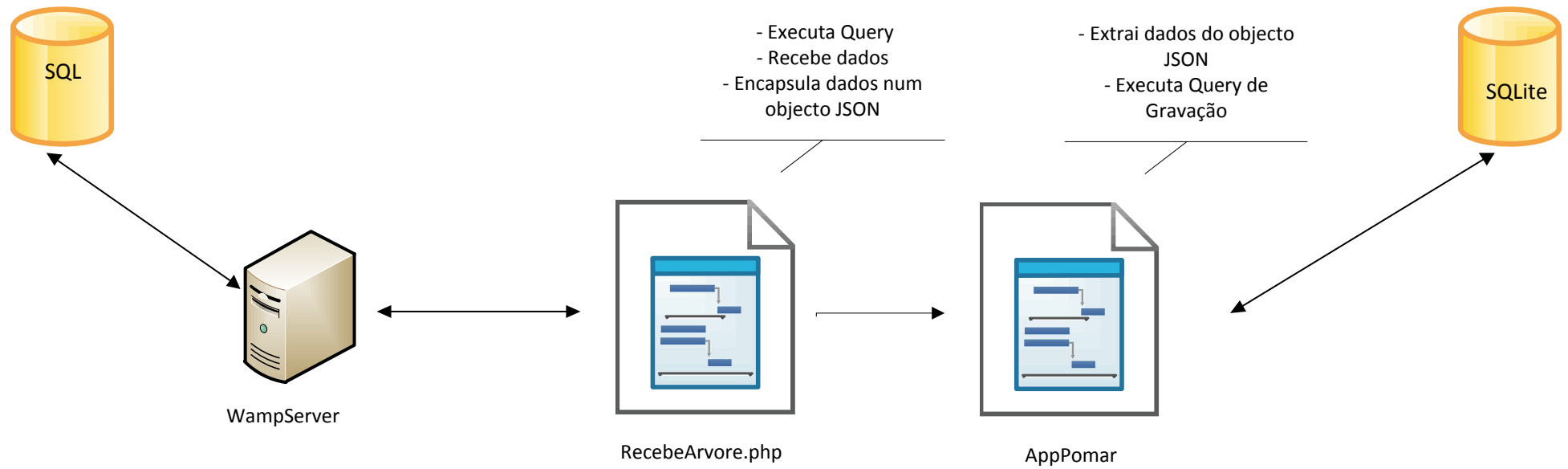



Ilustração 4 - Esquema da Sincronização

Aplicação Web

A aplicação web permite aos produtores visualizar e editar toda a informação recolhida pelo dispositivo móvel. É aqui que podem corrigir possíveis erros durante a recolha de dados e também onde podem ver graficamente os dados que já estão armazenados. Para aceder a este portal devem estar registados e está disponível no endereço: <http://pomardigital.net76.net>.



Faça o Login:

ID Produtor:

Password:

Ilustração 5 - HomePage da Aplicação Web

Pode ser editada informação dos dados pessoais, de cada árvore individualmente, das intervenções que foram feitas em cada árvore e visualizar graficamente todos estes dados para ter noção de como foi evoluindo o seu pomar.

Árvores

Produtor	Árvores	Intervenções	Gráficos	Sair
----------	---------	--------------	----------	------

Número Árvore:

Data Plantação:

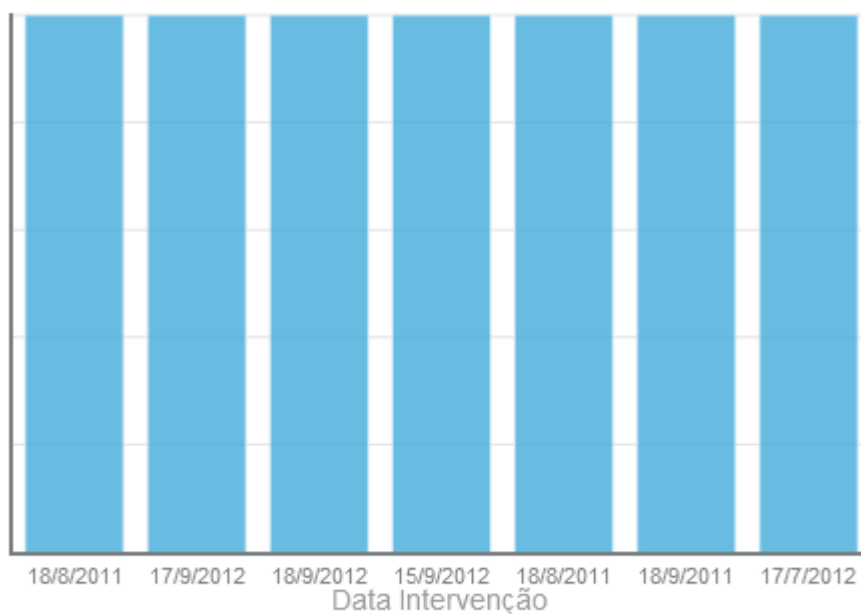
Data Implementação:

Tipo:

	Número Árvore	Data Plantação	Data Implementação	Tipo	
<input type="button" value="Editar"/>	14	15/7/2011	16/7/2011	Limoeiro	<input type="button" value="Apagar"/>
<input type="button" value="Editar"/>	11	16/8/2012	16/8/2012	Macieira	<input type="button" value="Apagar"/>
<input type="button" value="Editar"/>	12	16/8/2012	16/8/2012	Limoeiro	<input type="button" value="Apagar"/>

Ilustração 6 - Edição de Informação

Limpeza por árvore



Total de Limpezas = 7

[Voltar](#)

Ilustração 7- Gráfico de Limpezas

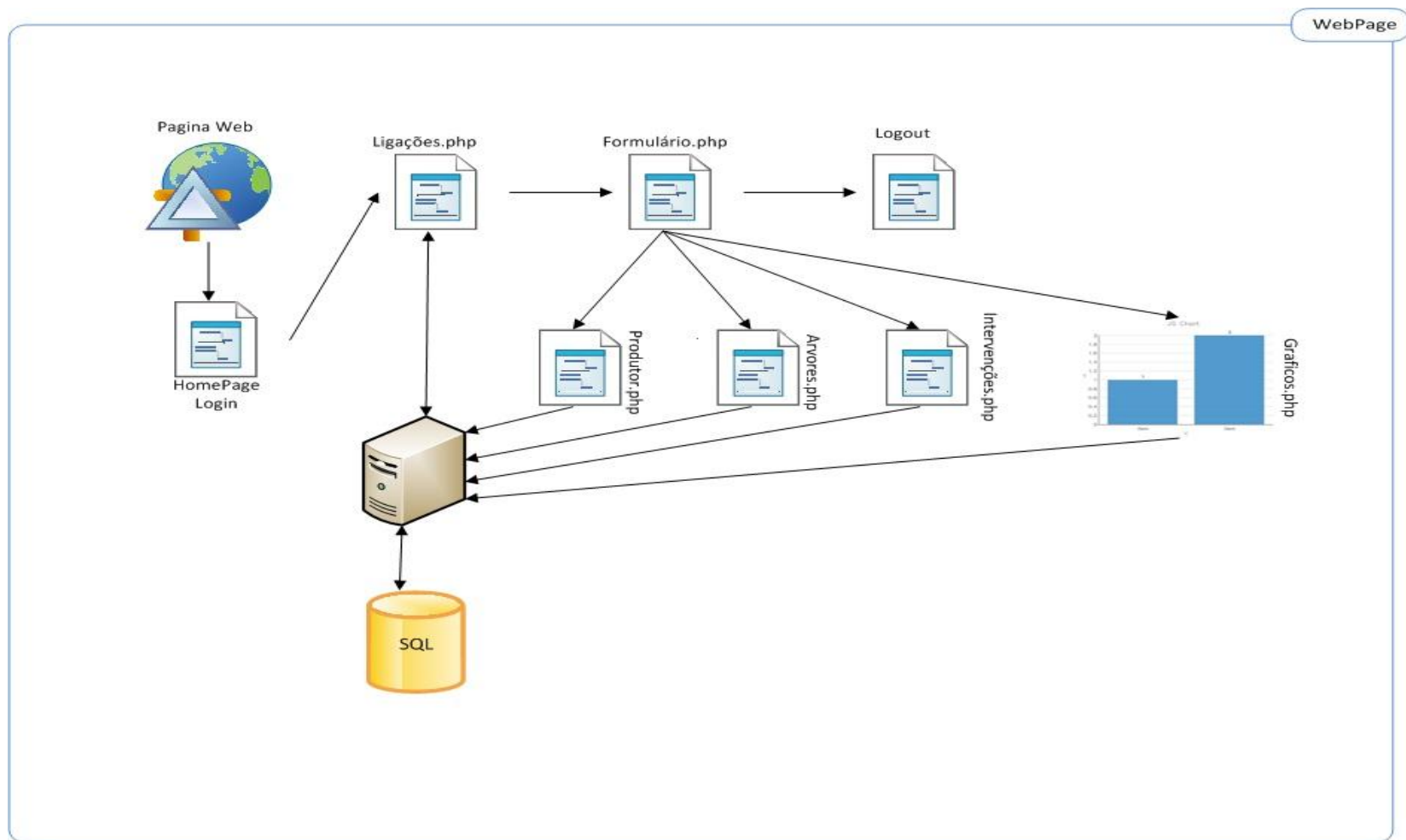


Ilustração 8 – EsquemaWebPage

JSCharts

Para a construção dos gráficos foi utilizado um gestor gráfico baseado em JavaScript chamado JS Charts. Este gestor permite criar de forma rápida e simples gráficos de todos os tipos. Para o utilizar basta utilizar um script que está disponível em <http://www.jscharts.com/>.

De seguida demonstramos como pode ser usado este gestor:

```
<html><head>
<title>Quantidade </title>
<!-- Utiliza o script JSCharts-->
<script type="text/javascript" src="jscharts.js"></script>
</head>
<body>
<?php
    $array = array();
    //Liga base de dados
    $conexao = mysql_connect('server','user','pass');
    mysql_select_db('basededados',$conexao);
    //Selec dos dados
    $sql="select Quantidade, Data_Intervencao from intervencoes
where Motivo = 'Colheita'";
    $resultado = mysql_query($sql) or die ("Erro
.: ".mysql_error());

    while($row=mysql_fetch_assoc($resultado))
    $array[]=$row;
    $total=0;
    //Cria do array de dados
    foreach ($array as $a){
        $total = $total + $a['Quantidade'];

    $arrayi[] =
    "[".$a['Data_Intervencao'].",".$a['Quantidade']."]";

    if (mysql_num_rows($resultado)>0)
    $s = implode(',',$arrayi);
    else {
        $erro='Sem Dados';
    }

?>
<font size="3" face="arial" color="66CC00">
<!-- Em caso de n array vazio-->
<?php echo $erro."\n";?>
<BR>
<BR>
</font>
<!-- Propriedades utilizadas no gráfico-->
<div id="graph">Loading graph...</div>
<script type="text/javascript">

    var myData = new Array(<?php echo $s ?>);
    var myChart = new JSChart('graph', 'bar');
    myChart.setDataArray(myData);
    myChart.setBarColor('#42aBdB');
    myChart.setBarOpacity(0.8);
```

```

myChart.setBarBorderColor('#D9EDF7');
myChart.setBarValues(false);
myChart.setTitleColor('#8C8383');
myChart.setAxisColor('#777E81');
myChart.setAxisNameX('Data Colheita');
myChart.setAxisNameY('Quantidade');
myChart.setTitle('Data colheita');
myChart.setAxisValuesColor('#777E81');
myChart.setSize(500, 350);
myChart.draw();
</script>
<font size="3" face="arial" color="66CC00">
Total do Pomar = <?php echo $total."\n";?> Kg
<BR>
<BR>
</font>
</div>
<FORM><INPUT TYPE="button" VALUE="Voltar" onClick="history.go(-
1);return true;"></FORM>
</body>
</html>

```

Conclusão

A aplicação cliente faz a integração Server Side de dados num banco-de-dados MySQL.

As linguagens Server Side permitem a criação de algoritmos, em PHP, úteis nas tarefas de gestão, processamento e análise de dados.

O processamento de dados é integrado numa plataforma Online designada R web. Esta plataforma permite efectuar complexas análises descritivas e inferenciais, conferindo, inclusivamente, a possibilidade de fazer forecasting.

A nível metodológico um script PHP gera, em função das variáveis em questão, um código fonte (linguagem matemática R) que será executado na plataforma R web e um CVS (Comma Separated Values) contendo os dados. Os resultados são graficamente interpretáveis e podem novamente ser importados para uma folha relatório.

Em relação às variáveis que possam influenciar os resultados do forecasting, podemos incluir diversas conforme o produtor em questão onde for implementada a aplicação. Por exemplo, com o avançar da idade da árvore, esta vai perdendo capacidade de produção; a qualidade da terra e o tratamento que lhe é dado também influenciam o crescimento e a posterior quantidade de frutos produzidos.

Para além das anteriores variáveis também podem ser consideradas as condições climáticas, as doenças que as árvores podem ter e as diversas intervenções a que podem ser sujeitas.

Todos estes dados foram impossíveis de quantificar, porque dependem do produtor e não nos foi possível fazer a implementação da mesma.

Bibliografia

- <http://developer.android.com/index.html>
- <http://www.informaticon.com.br/j/>
- <http://www.vogella.com/articles/AndroidPerformance/article.html>
- <http://www.w3schools.com/>
- <http://www.php.net/>
- <http://www.jscharts.com/>
- <http://stackoverflow.com/>
- <http://code.google.com/p/zxing/>

Anexos

Configuração da biblioteca ZXIng para utilização do QR CODE

Anexo 1 - Biblioteca Zxing

Para que o scanner funcione na nossa aplicação sem ser necessário recorrer a aplicações externas, ou seja, uma aplicação standalone, é necessário usar a biblioteca Zxing, já descrita anteriormente.

Esta biblioteca está disponível para download no site <http://code.google.com/p/zxing/downloads/detail?name=ZXing-2.0.zip>. Depois de fazer o download e o descompactamento da pasta, criámos um novo projecto no Eclipse e importámos a pasta “android”.

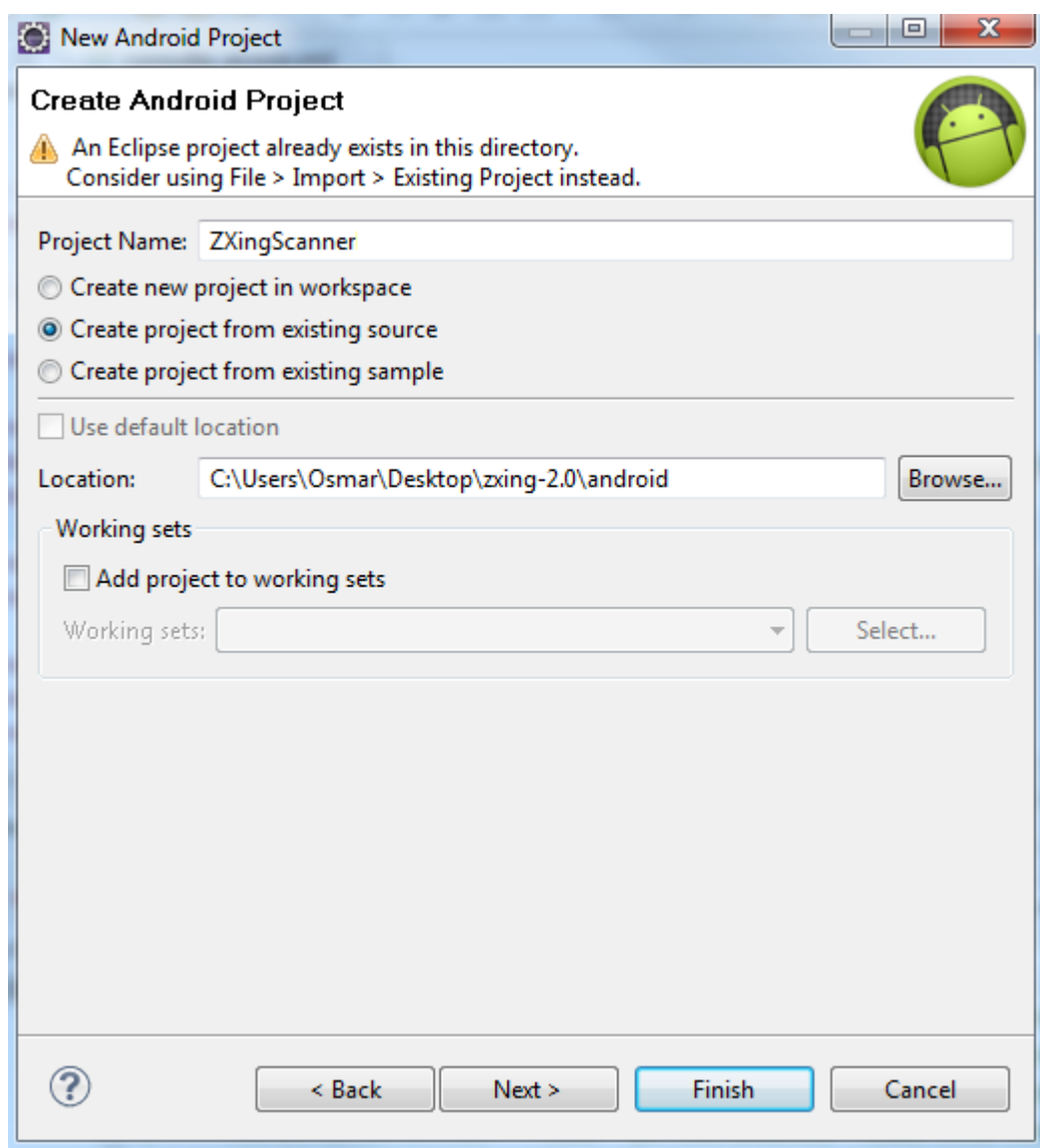


Ilustração 9 - ZXing - Novo Projecto

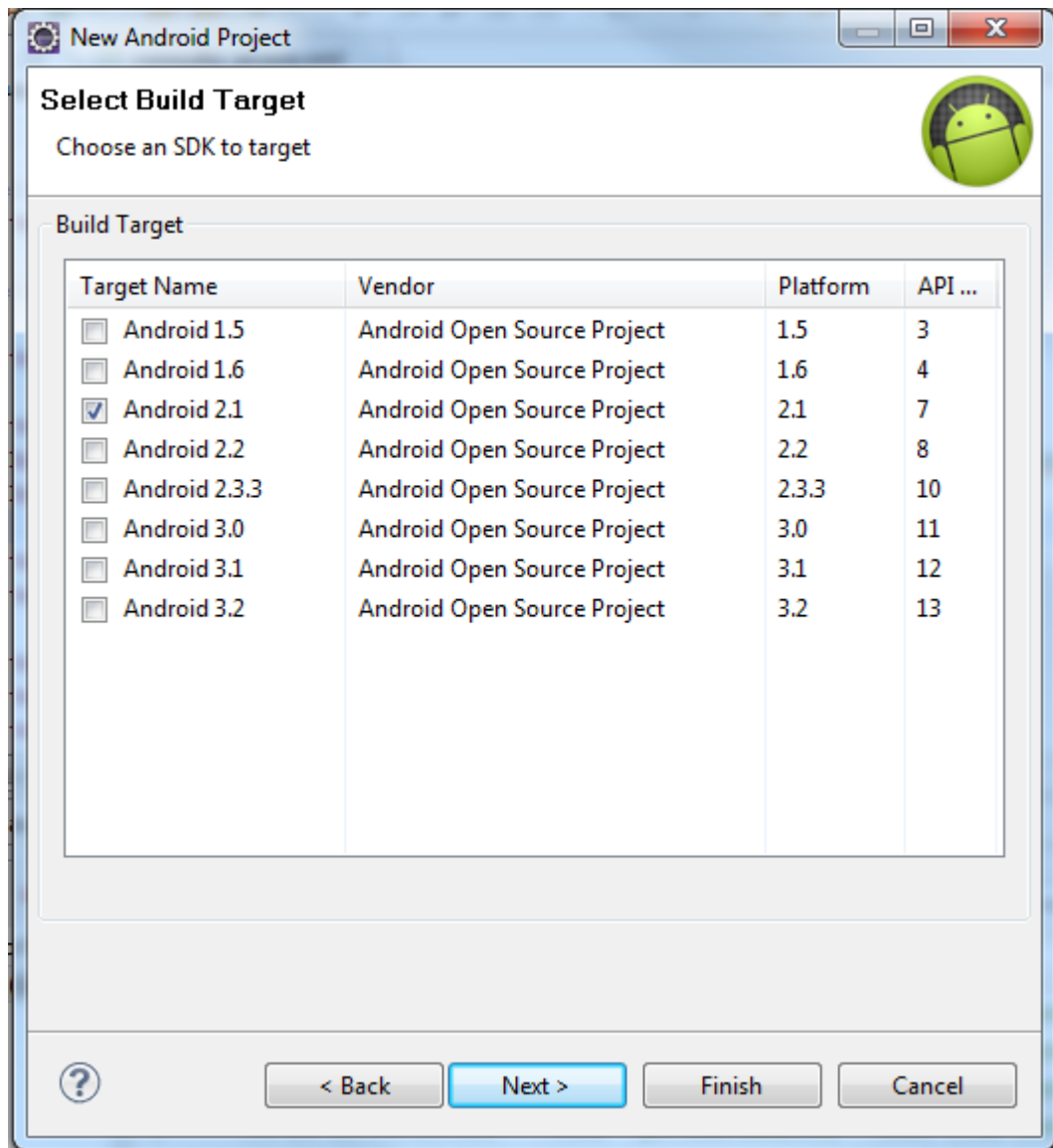


Ilustração 10 - ZXing - Target

Depois disto, é necessário aceder às propriedades do projeto: clicar com o botão do lado direito do rato em cima do projecto → Propriedades. Selecionar “Android” e colocar um pisco em “Is Library”.

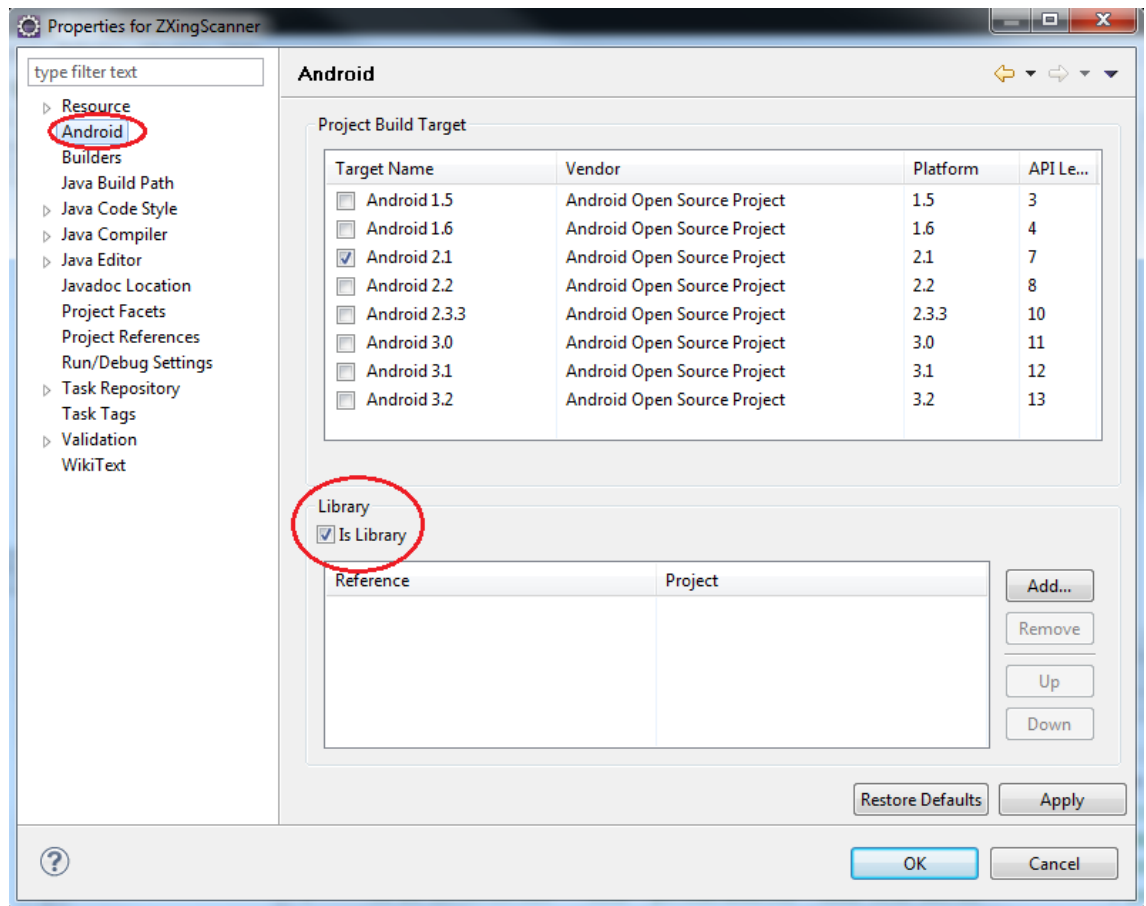


Ilustração 11 - ZXing Properties

De seguida, seleccionar “Java Build Path” e “Add External JARs...”, para adicionar o ficheiro “core.jar” que se encontra na pasta “core” dentro da pasta descompactada anteriormente.

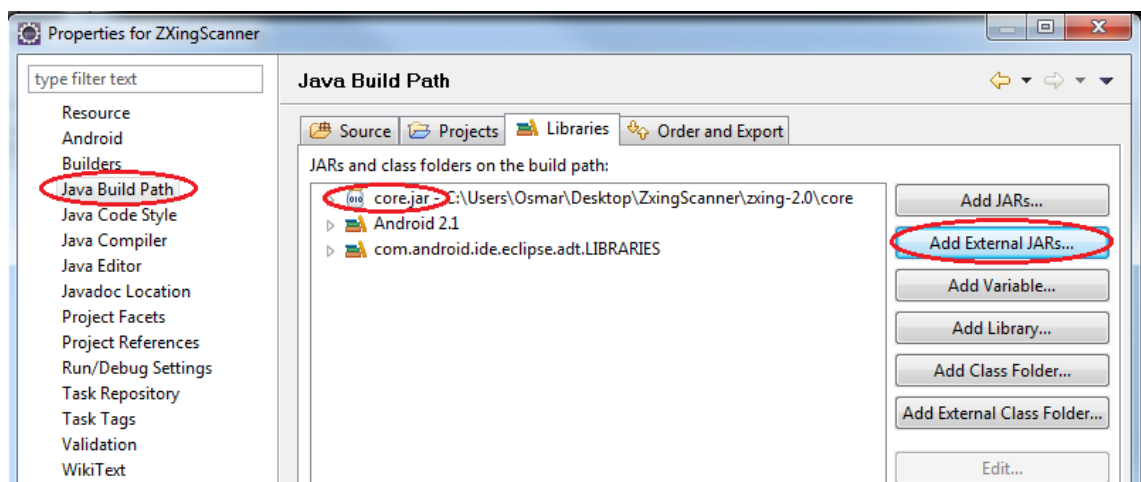


Ilustração 12 - ZXing - Java Build Path

Agora que temos o nosso projecto Zxing criado e pronto a usar, é necessário incluí-lo no nosso projecto principal. No nosso projecto principal: clicar com o botão do lado direito do rato em cima do projecto → Propriedades. Selecionar “Android” e “Add”. O projecto Zxing vai aparecer numa janela; é só adicionar.

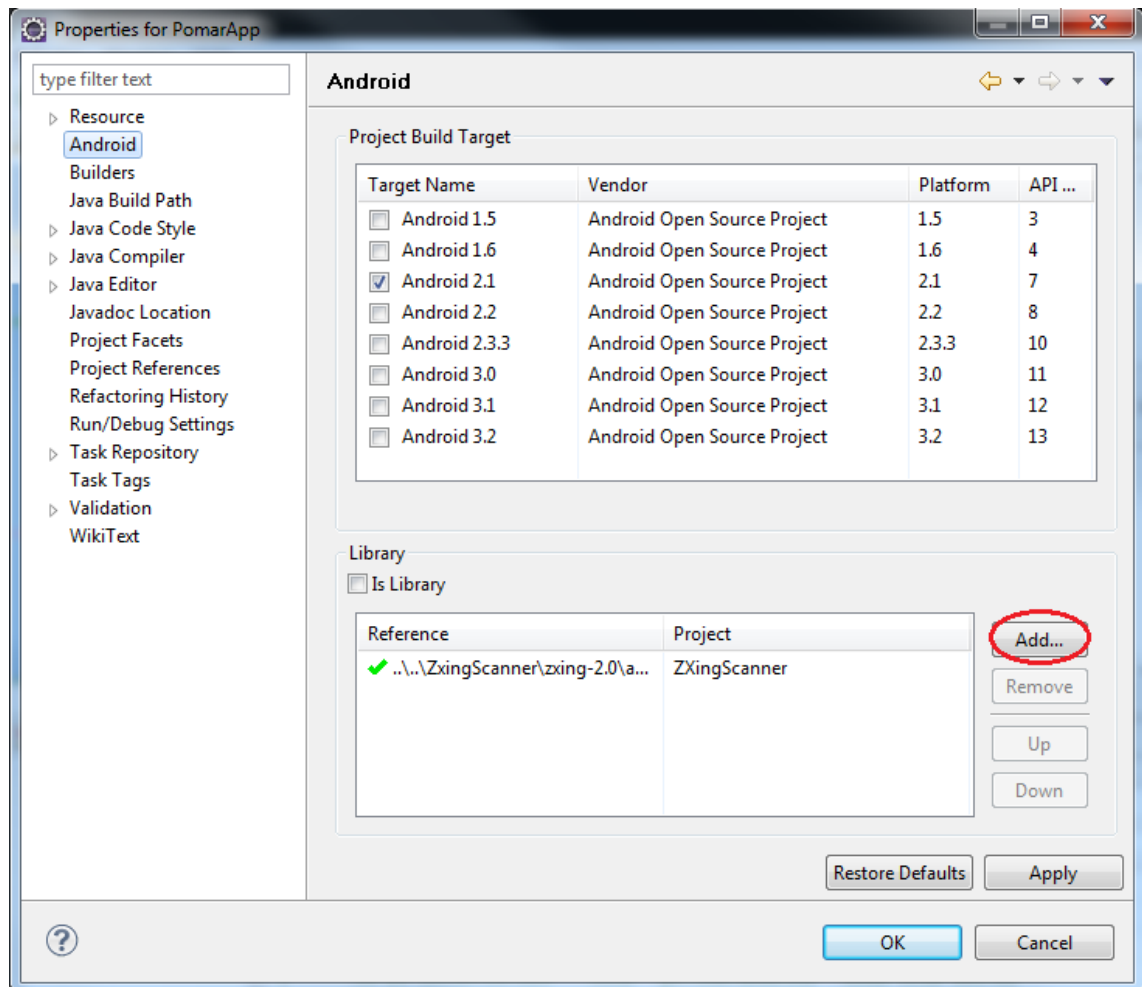


Ilustração 13 - ZXing Resume

Para finalizar, temos que incluir também o ficheiro “core.jar” da mesma forma como incluímos no projecto Zxing.

Agora vem o código: primeiro vamos adicionar o código xml ao “AndroidManifest.xml”:

```
<activity
    android:name="com.google.zxing.client.android.CaptureActivity"
    android:screenOrientation="landscape"

    android:configChanges="orientation|keyboardHidden"

    android:theme="@android:style/Theme.NoTitleBar.Fullscreen"

    android:windowSoftInputMode="stateAlwaysHidden">
    <intent-filter>
```

```

        <action android:name="android.intent.action.MAIN"/>
        <category
android:name="android.intent.category.DEFAULT"/>
        </intent-filter>
        <intent-filter>
        <action
android:name="com.google.zxing.client.android.SCAN"/>
        <category
android:name="android.intent.category.DEFAULT"/>
        </intent-filter>

    </activity>

```

Esta actividade vai ser a responsável pelo scanner.

No código Java, temos que ter uma função que faça a chamada desta actividade:

```

public void chamascanner() {

    Intent intent= new Intent("com.google.zxing.client.android.SCAN");
        intent.putExtra("SCAN_MODE", "QR_CODE_MODE");
        startActivityForResult(intent,0);
    //O Intent é uma abstração descritiva da acção a ser executada,
    faz basicamente uma chamada a uma "actividade" presente noutro
    projecto, ou seja, executa código presente noutro projecto.
    }

@Override
    public void onActivityResult(int requestCode, int resultCode, Intent
intent)
    {
    //É aqui que a magia acontece... Nesta "actividade" recebemos o
    resultado do scanner e o manipulamos como quisermos.
        if(requestCode==0)
        {
            if(resultCode==RESULT_OK)
            {
                String contents=intent.getStringExtra("SCAN_RESULT");
                idArvore=contents.toString();
                //Falta código, isto é só um exemplo. O resto estará em
                anexo.
            }
            else if(resultCode==RESULT_CANCELED)
            {
            }
        }
    }}

```

Assim, é só usar estas duas funções e temos a nossa aplicação com scanner standalone.

Source Code - Aplicação Web

Anexo 2 – HomePage

[illegible]

Anexo 3 – Ligações

```

<?php
if( !session_id() ) session_start();
if( isset($_POST['nome']) ) {
    //recebe por POST dados do produtor
    $nome = addslashes($_POST['nome']);
    $passe = addslashes($_POST['senha']);
    //conecta base de dados
    $bd = new
PDO("mysql:host=mysql2.000webhost.com;dbname=a3479280_pomar","a3479280
_root", "ulht2012");
    //seleciona o produtor, se existir
    $query = "SELECT * FROM produtor " .
        "WHERE idProdutor='$nome' AND password='$passe'";
    $result = $bd->query($query);
    if($result->fetch()) {
        //inicia a variável de sessão
        $_SESSION['nome'] = $nome;
    }
}
if( isset($_SESSION['nome']) ) {
    header("Location: Formulario.php");
    return;
}
// em caso de login errado reencaminha para a página
else header("Location: Logout.php");
?>

```

Anexo 4 – Formulário Principal

```
<html><head>
<title>Pomar</title>
<?php
//inclusão dos scrpits de formatação de alteração da data e do menu
comum a todas as páginas
include 'Scripts.php';
?>
</head>
<span style="padding-left:120px">

</span>
<br>

<font size="4" face="arial" color="66CC00">
<br>
<b>Formulário de dados do Pomar:</b>
</font>
<br>
<br>
<!--Ligações dos botoes do menu aos respectivos php-->
<ul id="jsddm">
<li><a href="Produtor.php">Produtor</a>

<li><a href="Arvore.php">Árvores</a>
</li>
<li><a href="Intervencoes.php">Intervenções</a>
<li><a href="Gr&aacute;ficos">Gr&aacute;ficos</a>
</li>
<li><a href="Graficos.php">Colheitas/Data</a></li>
<li><a href="Graficos2.php">Limpezas/Data</a></li>
<li><a href="Graficos3.php">Tratamento/Data</a></li>
<li><a href="Graficos4.php">Outros Motivos/Data</a></li>
</ul>
<li><a href="logout.php">Sair</a>
</ul>
<br>
<br>
<br>
<br>
<span style="padding-left:60px">

</font>
</body>
</html>
```

Anexo 5 – Scripts

```
<!--jquery-DD-Menu-Starts-->
<style type="text/css">
/* menu styles */
#jsddm
{ margin: 0;
padding: 0;}

#jsddm li
{ float: left;
list-style: none;
font: 12px Tahoma, Arial;}

#jsddm li a
{ display: block;
background: #324143; /*DARK-GREEN*/
padding: 5px 12px;
text-decoration: none;
border-right: 1px solid white;
width: 70px;
color: #EAFED;
white-space: nowrap;}

#jsddm li a:hover
{ background: #24313C;}

#jsddm li ul
{ margin: 0;
padding: 0;
position: absolute;
visibility: hidden;
border-top: 1px solid white;}

#jsddm li ul li
{ float: none;
display: inline;}

#jsddm li ul li a
{ width: auto;
background: #A9C251; /*LIGHT-GREEN*/
color: #24313C;}

#jsddm li ul li a:hover
{ background: #8EA344;}
</style>

<!--Menu comum a todas as páginas-->
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.2.6/jquery.min.js"
type="text/javascript"></script>
<script type="text/javascript">
var timeout = 500;
var closetimer = 0;
var ddmenuitem = 0;

function jsddm_open()
{ jsddm_canceltimer();
jsddm_close();
```

```

ddmenuitem = $(this).find('ul').eq(0).css('visibility', 'visible');}

function jsddm_close()
{ if(ddmenuitem) ddmenuitem.css('visibility', 'hidden');}

function jsddm_timer()
{ closetimer = window.setTimeout(jsddm_close, timeout);}

function jsddm_canceltimer()
{ if(closetimer)
{ window.clearTimeout(closetimer);
closetimer = null;}}

$(document).ready(function()
{ $('#jsddm > li').bind('mouseover', jsddm_open);
$('#jsddm > li').bind('mouseout', jsddm_timer);});

document.onclick = jsddm_close;
</script>
<!--formatacao da data-->
<script language="JavaScript">

function DataHora(evento, objeto){
    var keypress=(window.event)?event.keyCode:evento.which;
    campo = eval (objeto);
    if (campo.value == '00/00/0000')
    {
        campo.value=""
    }

    caracteres = '0123456789';
    separacao1 = '/';
    separacao2 = ' ';
    separacao3 = ':';
    conjunto1 = 2;
    conjunto2 = 5;
    conjunto3 = 10;
    conjunto4 = 13;
    conjunto5 = 16;
    if ((caracteres.search(String.fromCharCode (keypress))!=-1) &&
campo.value.length < (10))
    {
        if (campo.value.length == conjunto1 )
            campo.value = campo.value + separacao1;
        else if (campo.value.length == conjunto2)
            campo.value = campo.value + separacao1;
        else if (campo.value.length == conjunto3)
            campo.value = campo.value + separacao2;

    }
    else
        event.returnValue = false;
}
</script>

```

Anexo 6 – Produtor

```
<?php
//ligacao ao php que faz as operacoes na BD
include 'sqlprodutor.php';
?>
<html>
<title>Produtor</title>
<head>
<script type="text/javascript" src="Validacao.js"></script>
<span style="padding-left:90px">

</span>
<font size="4" face="arial" color="A7C942">
<p>
Produtor
<font>
<br>
<br>
<style type="text/css">
#apDiv1 {
    position:absolute;
    width:362px;
    height:295px;
    z-index:1;
    left: 555px;
    top: 63px;
}

#intervencoes
{
font-family:"Trebuchet MS", Arial, Helvetica, sans-serif;
width:100%;
border-collapse:collapse;
}
#intervencoes td, #intervencoes th
{
font-size:1em;
border:1px solid #98bf21;
padding:3px 7px 2px 7px;
}
#intervencoes th
{
font-size:1.1em;
text-align:left;
padding-top:5px;
padding-bottom:4px;
background-color:#A7C942;
color:#ffffff;
}
#intervencoes tr.alt td
{
color:#000000;
background-color:#EAF2D3;
}
</style>
<?php
include 'Scripts.php';
?>
```

```

<div>
<!--Menu-->
<ul id="jsddm">
<li><a href="Produtor.php">Produtor</a>
<li><a href="Arvore.php">Arvores</a>
</li>
<li><a href="Intervencoes.php">Intervenções</a>
<li><a href="Graficos.php">Graficos</a>
</li>
<li><a href="Graficos2.php">Colheitas/Data</a></li>
<li><a href="Graficos2.php">Limpezas/Data</a></li>
<li><a href="Graficos3.php">Tratamento/Data</a></li>
<li><a href="Graficos4.php">Outros Motivos/Data</a></li>
</li>
<li><a href="logout.php">Sair</a>
</li>
</ul>
</div>
</head>
<br>
<br>
<style type="text/css">
table { border-collapse: collapse; }
table, th, td { border: 1px solid black; }
th { background-color: 66CC00; color: white; }
</style>
<body>
<!--Tabela de leitura dos dados-->
<?php if(isset($_GET['insere'])) && $_GET['insere']==1 ||
isset($_GET['upd'])) { ?>
<form action="" method="post">
<input type="hidden" name="idProdutor" value="<?php echo
isset($_GET['upd']) ? $_GET['upd']: 0;?>" />
Nome: <span style="padding-left:50px"><input type="text" name="Nome"
value="<?php echo isset($_GET['upd']) ? $linha['Nome']: '';?>" /><br/>
</span>
Morada:<span style="padding-left:36px"> <input type="text"
name="Morada" value="<?php echo isset($_GET['upd']) ?
$linha['Morada']: '';?>" /><br/>
</span>
Password:<span style="padding-left:18px"> <input type="text"
name="password" value="<?php echo isset($_GET['upd']) ?
$linha['password']: '';?>" /><br/>
<input onclick="validar(this)" type="submit" value="<?php echo
isset($_GET['upd']) ? 'Atualizar': 'Inserir';?>" />
<button onclick="action='Produtor.php'">Cancelar</button>
</span>
</form>
<?php } ?>
<!--Tabela de edição dos dados-->
<table id="intervencoes">
<tr>
<th></th>
<th>Nome</th>
<th>Morada</th>
<th>Password</th>
</tr>
</tr>
<?php
$result = $bd->query("SELECT * FROM produtor WHERE idProdutor=
".$_SESSION['nome']); //nome é id produtor inserido no login
while($linha = $result->fetch()) {
    echo "<tr>";

```



```
        echo "<td align='center'><a href='?upd={\$linha['idProdutor']}'>
<input type=button value=Editar> </a></td>";
        echo "<td>" . \$linha['Nome'] . "</td>";
        echo "<td>" . \$linha['Morada'] . "</td>";
        echo "<td>" . \$linha['password'] . "</td>";
        echo "</tr>";
    }
?>
</table>
</body>
</html>
```

Anexo 7 – SQLProdutor

```
<?php
if( !session_id() ) session_start();
//Apagar dados
$bd = new
PDO("mysql:host=mysql2.000webhost.com;dbname=a3479280_pomar","a3479280
_root", "ulht2012");
if(isset($_GET['del'])) {
    $id = $_GET['del'];
    $query = "DELETE FROM produtor WHERE idProdutor=$id";
    $bd->query($query);
}
//Inserir dados
if(isset($_POST['idProdutor']) && $_POST['idProdutor']!=0) {
    $Nome = $_POST['Nome'];
    $Morada = $_POST['Morada'];
    $password = $_POST['password'];
    $query = "INSERT INTO produtor (idProdutor, Nome, Morada, password)
    ".
        "VALUES (NULL, '$Nome', '$Morada', '$password')";
    $bd->query($query);
}
//Update de dados
if(isset($_POST['idProdutor']) && $_POST['idProdutor']!=0) {
    $id = $_POST['idProdutor'];
    $Nome = $_POST['Nome'];
    $Morada = $_POST['Morada'];
    $password = $_POST['password'];
    $query = "UPDATE produtor SET " .
        "Nome='$Nome', Morada='$Morada' " .
        "WHERE idProdutor=$id";
    $bd->query($query);
}
//Selec機 de dados
if(isset($_GET['upd'])) {
    $query = "SELECT * FROM produtor WHERE idProdutor=".$_GET['upd'];
    $result = $bd->query($query);
    $linha = $result->fetch();
}
?>
```

Anexo 8 – Árvores

```
<?php
include 'sqlarvores.php';
?>
<html>
<title>rvores</title>
<head>
<?php
include 'Scripts.php';
?>
<span style="padding-left:90px">

</span>
<font size="4" face="arial" color="A7C942">
<p>
rvores
<font>
<br>
<br>
<style type="text/css">
#apDiv1 {
    position:absolute;
    width:362px;
    height:295px;
    z-index:1;
    left: 555px;
    top: 63px;
}

#intervencoes
{
font-family:"Trebuchet MS", Arial, Helvetica, sans-serif;
width:100%;
border-collapse:collapse;
}
#intervencoes td, #intervencoes th
{
font-size:1em;
border:1px solid #98bf21;
padding:3px 7px 2px 7px;
}
#intervencoes th
{
font-size:1.1em;
text-align:left;
padding-top:5px;
padding-bottom:4px;
background-color:#A7C942;
color:#ffffff;
}
#intervencoes tr.alt td
{
color:#000000;
background-color:#EAF2D3;
}
</style>
<?php
include 'Scripts.php';
```

```

?>
<script type="text/javascript">
//Função para confirmar eliminaç  de dados
function confirmar(id)
{
var r=confirm("Eliminar rvore?")
if (r==true)
{

window.location.href = "RemoverArvores.php?del="+id;

}
else
{
window.location="Arvore.php"
}
}
</script>
<script language="javascript" src="Validacao.js"></script>
<div>
<ul id="jsddm">
<li><a href="Produtor.php">Produtor</a>
<li><a href="Arvore.php">rvores</a>
</li>
<li><a href="Intervencoes.php">Interven es</a>
<li><a href="Gr ficos">Gr ficos</a>
</li>
<li><a href="Graficos.php">Colheitas/Data</a></li>
<li><a href="Graficos2.php">Limpezas/Data</a></li>
<li><a href="Graficos3.php">Tratamento/Data</a></li>
<li><a href="Graficos4.php">Outros Motivos/Data</a></li>
</ul>
<li><a href="logout.php">Sair</a>
</div>
</head>
<br>
<br>
<style type="text/css">
table { border-collapse: collapse; }
table, th, td { border: 1px solid black; }
th { background-color: 66CC00; color: white; }
</style>
<body>
<?php if(isset($_GET['insere']) && $_GET['insere']==1 ||
isset($_GET['upd'])) { ?>
<form action="" method="post" onSubmit="return validar(this)">
<input type="hidden" name="IDArvores" value="<?php echo
isset($_GET['upd']) ? $_GET['upd']: 0;?>" />
Nmero rvore: <span style="padding-left:45px"><input type="text"
name="numArvore" value="<?php echo isset($_GET['upd']) ?
$linha['numArvore']: '';?>" /><br/>
</span>
Data Planta : <span style="padding-left:43px"><input type="text"
name="Data_Plantacao" onKeyPress="DataHora(event, this)" value="<?php
echo isset($_GET['upd']) ? $linha['Data_Plantacao']: '';?>" /><br/>
</span>
Data Implementa : <span style="padding-left:8px"><input type="text"
name="Data_Implementacao" onKeyPress="DataHora(event, this)"
value="<?php echo isset($_GET['upd']) ? $linha['Data_Implementacao']:
'';?>" /><br/>
</span>

```

```

Tipo:<span style="padding-left:135px"><input type="text"
name="Tipo_Arvore" value="<?php echo isset($_GET['upd']) ?
$linha['Tipo_Arvore']: '';?>" /><br/>
</span>
<input type="submit" value="<?php echo isset($_GET['upd']) ?
'Atualizar': 'Inserir';?>"/>
<button onclick="action='Produtor.php'">Cancelar</button>
</form>
<?php } ?>
<table id="intervencoes">
<tr>
<th></th>
<th>Numero rvore</th>
<th>Data Planta機</th>
<th>Data Implementa機</th>
<th>Tipo</th>
<th></th>
</tr>
<?php
$result = $bd->query("SELECT * FROM arvores WHERE Produtor_idProdutor=
".$_SESSION['nome']." AND inativo='0'");
while($linha = $result->fetch()) {
    echo "<tr>";
    echo "<td align='center'><a href='?upd={$linha['IDArvores']}'>
<INPUT TYPE=button Value=Editar onClick=alert('ok')></a></td>";
    echo "<td>" . $linha['numArvore'] . "</td>";
    echo "<td>" . $linha['Data_Plantacao'] . "</td>";
    echo "<td>" . $linha['Data_Implementacao'] . "</td>";
    echo "<td>" . $linha['Tipo_Arvore'] . "</td>";
    echo "<td align='center'> <input type=button
onclick=confirmar({$linha['numArvore']}) value=Apagar /></a></td>";
    echo "</tr>";
}
?>
</table>
</body>
</html>

```

Anexo 9 – SQLArvores

```
<?php
if( !session_id() ) session_start();

$bd = new
PDO("mysql:host=mysql2.000webhost.com;dbname=a3479280_pomar","a3479280
_root", "ulht2012");
if(isset($_GET['del'])) {
    $id = $_GET['del'];
    $query = "DELETE FROM arvores WHERE IDArvores=$id";
    $bd->query($query);
}
if(isset($_POST['IDArvores']) && $_POST['IDArvores']==0) {
    $numArvore = $_POST['numArvore'];
    $Data_Plantacao = $_POST['Data_Plantacao'];
    $Data_Implementacao = $_POST['Data_Implementacao'];
    $Tipo_Arvore = $_POST['Tipo_Arvore'];
    $query = "INSERT INTO arvores
(IDArvores,numArvore,Data_Plantacao,Data_Implementacao,Tipo_Arvore) ".
        "VALUES (NULL, '$numArvore',
'$Data_Plantacao','$Data_Implementacao','$Tipo_Arvore')";
    $bd->query($query);
}
if(isset($_POST['IDArvores']) && $_POST['IDArvores']!=0) {
    $id = $_POST['IDArvores'];
    $numArvore = $_POST['numArvore'];
    $Data_Plantacao = $_POST['Data_Plantacao'];
    $Data_Implementacao = $_POST['Data_Implementacao'];
    $Tipo_Arvore = $_POST['Tipo_Arvore'];
    $query = "UPDATE arvores SET " .
        "numArvore='$numArvore',Data_Plantacao='$Data_Plantacao',Data_Implemen
        tacao='$Data_Implementacao',Tipo_Arvore='$Tipo_Arvore',estado='1' " .
        "WHERE IDArvores=$id";
    $bd->query($query);
}
if(isset($_GET['upd'])) {
    $query = "SELECT * FROM arvores WHERE IDArvores=" . $_GET['upd'];
    $result = $bd->query($query);
    $linha = $result->fetch();
}
?>
```

Anexo 10 – Intervenções

```
<?php
include 'sqlintervencoes.php';
?>
<html>
<head>
<title>Intervenções</title>
<?php
include 'Scripts.php';
?>

<span style="padding-left:90px">

</span>
<font size="4" face="arial" color="66CC00">
<p>
Intervenções
<font>
<br>
<br>
<style type="text/css">
#apDiv1 {
    position:absolute;
    width:362px;
    height:295px;
    z-index:1;
    left: 555px;
    top: 63px;
}

#intervencoes
{
font-family:"Trebuchet MS", Arial, Helvetica, sans-serif;
width:100%;
border-collapse:collapse;
}
#intervencoes td, #intervencoes th
{
font-size:1em;
border:1px solid #98bf21;
padding:3px 7px 2px 7px;
}
#intervencoes th
{
font-size:1.1em;
text-align:left;
padding-top:5px;
padding-bottom:4px;
background-color:#A7C942;
color:#ffffff;
}
#intervencoes tr.alt td
{
color:#000000;
background-color:#EAF2D3;
}
</style>
```

```

<?php
include 'Scripts.php';
?>

<script type="text/javascript">
function confirmar(id)
{
var r=confirm("Eliminar Intervenção?")
if (r==true)
{

    window.location.href = "RemoverIntervencoes.php?del="+id;

}
else
{
    window.location="Intervencoes.php"
}
}
</script>

<div>
<ul id="jsddm">
<li><a href="Produtor.php">Produtor</a>
<li><a href="Arvore.php">Arvores</a>
</li>
<li><a href="Intervencoes.php">Intervenções</a>
<li><a href="Graficos.php">Graficos</a>
<ul>
<li><a href="Graficos.php">Colheitas/Data</a></li>
<li><a href="Graficos2.php">Limpezas/Data</a></li>
<li><a href="Graficos3.php">Tratamento/Data</a></li>
<li><a href="Graficos4.php">Outros Motivos/Data</a></li>
</ul>
<li><a href="logout.php">Sair</a>
</div>
</head>
<br>
<br>
<style type="text/css">
table { border-collapse: collapse; }
table, th, td { border: 1px solid black; }
th { background-color: 66CC00; color: white; }
</style>
<body>
<?php if(isset($_GET['insere']) && $_GET['insere']==1 ||
isset($_GET['upd'])) { ?>
<form action="" method="post">
<input type="hidden" name="IDIntervencoes" value="<?php echo
isset($_GET['upd']) ? $_GET['upd']: 0;?>" />
Data Intervenção:<span style="padding-left:1px"> <input type="text"
name="Data_Intervencao" onKeyPress="DataHora(event, this)"
value="<?php echo isset($_GET['upd']) ? $linha['Data_Intervencao']:
'';?>" /><br/>
</span>
Motivo:<span style="padding-left:89px"><input type="text"
name="Motivo" value="<?php echo isset($_GET['upd']) ?
$linha['Motivo']: '';?>" /><br/>
</span>

```



```

Quantidade:<span style="padding-left:49px"><input type="text"
name="Quantidade" value="<?php echo isset($_GET['upd']) ?
$linha['Quantidade']: '';?>" />Kg<br/>
</span>
Observa  s:<span style="padding-left:36px"><input type="text"
name="observacao" value="<?php echo isset($_GET['upd']) ?
$linha['observacao']: '';?>" /><br/>
</span>
<input type="submit" value="<?php echo isset($_GET['upd']) ?
'Atualizar': 'Inserir';?>" />
<button onclick="action='Intervencoes.php'">Cancelar</button>
</form>
<?php } ?>
<table id="intervencoes">
<tr>
<th></th>
<th>Nmero rvore</th>
<th>Data Interven  </th>
<th>Motivo</th>
<th>Quantidade</th>
<th>Observa  s</th>
<th></th>
</tr>
<?php
$result = $bd->query("SELECT * FROM intervencoes WHERE
Arvores_Produtor_idProdutor= ".$_SESSION['nome']." AND inativo='0'");
while($linha = $result->fetch()) {
    echo "<tr>";
    echo "<td align='center'><a
href='?upd={$linha['IDIntervencoes']}'> <input type=button
value=Editar> </a></td>";
    echo "<td>" . $linha['Arvores_idArvores'] . "</td>";
    echo "<td>" . $linha['Data_Intervencao'] . "</td>";
    echo "<td>" . $linha['Motivo'] . "</td>";
    echo "<td>" . $linha['Quantidade'] . "</td>";
    echo "<td>" . $linha['observacao'] . "</td>";
    echo "<td align='center'> <input type=button
onclick=confirmar({$linha['IDIntervencoes']}) value=Apagar
/></a></td>";
    echo "</tr>";
}
?>
</table>
</body>
</html>

```

Anexo 11 – SQLIntervencoes

```
<?php
if( !session_id() ) session_start();

$bd = new
PDO("mysql:host=mysql2.000webhost.com;dbname=a3479280_pomar","a3479280
_root", "ulht2012");
if(isset($_GET['del'])) {
    $id = $_GET['del'];
    $query = "DELETE FROM intervencoes WHERE IDIntervencoes=$id";
    $bd->query($query);
}
if(isset($_POST['IDIntervencoes']) && $_POST['IDIntervencoes']==0) {
    $Data_Intervencao = $_POST['Data_Intervencao'];
    $Motivo = $_POST['Motivo'];
    $query = "INSERT INTO intervencoes
(IDIntervencoes,Data_Intervencao,Motivo) ".
        "VALUES (NULL,'$Data_Intervencao','$Motivo')";
    $bd->query($query);
}
if(isset($_POST['IDIntervencoes']) && $_POST['IDIntervencoes']!=0) {
    $id = $_POST['IDIntervencoes'];
    $Data_Intervencao = $_POST['Data_Intervencao'];
    $Motivo = $_POST['Motivo'];
    $Quantidade = $_POST['Quantidade'];
    $observacao = $_POST['observacao'];
    $query = "UPDATE intervencoes SET " .

    "Data_Intervencao='$Data_Intervencao',Motivo='$Motivo',Quantidade='$Qu
antidade',observacao='$observacao',estado='1'" .
        "WHERE IDIntervencoes=$id";
    $bd->query($query);
}
if(isset($_GET['upd'])) {
    $query = "SELECT * FROM intervencoes WHERE IDIntervencoes=" .
$_GET['upd'];
    $result = $bd->query($query);
    $linha = $result->fetch();
}
?>
```

Anexo 12 – Gráfico Colheitas/Data

```
<html><head>
<title>Quantidade </title>
<script type="text/javascript" src="jscharts.js"></script>
</head>
<body>
<?php
    $array = array();
    //Liga 機 base de dados
    $conexao =
mysql_connect('mysql2.000webhost.com','a3479280_root','ulht2012');
    mysql_select_db('a3479280_pomar',$conexao);
    //selec 機 dos dados
    $sql="select Quantidade, Data_Intervencao from intervencoes
where Motivo = 'Colheita'";
    $resultado = mysql_query($sql) or die ("Erro
.: ".mysql_error());

    while($row=mysql_fetch_assoc($resultado))
    $array[]=$row;
    $total=0;
    //contador do total acumulado
    foreach ($array as $a){
        $total = $total + $a['Quantidade'];
    }
    //criacao do array para constru 機 do gráfico
    $arrayi[] =
    "[".$a['Data Intervencao'].",".$a['Quantidade']."]";

    if (mysql_num_rows($resultado)>0)
    $s = implode (',', $arrayi);
    else { //caso n 機 tenha dados registados
        $erro='Sem Dados';
    }

?>
<font size="3" face="arial" color="66CC00">
<?php echo $erro."\n";?>
<BR>
<BR>
</font>
<!-- Constru 機 do gráfico-->
<div id="graph">Loading graph...</div>
<script type="text/javascript">

    var myData = new Array(<?php echo $s ?>);
    var myChart = new JSChart('graph', 'bar');
    myChart.setDataArray(myData);
    myChart.setBarColor('#42aBdB');
    myChart.setBarOpacity(0.8);
    myChart.setBarBorderColor('#D9EDF7');
    myChart.setBarValues(false);
    myChart.setTitleColor('#8C8383');
    myChart.setAxisColor('#777E81');
    myChart.setAxisNameX('Data Colheita');
    myChart.setAxisNameY('Quantidade');
    myChart.setTitle('Data colheita');
    myChart.setAxisValuesColor('#777E81');
```

```
        myChart.setSize(500, 350);
        myChart.draw();
    </script>
    <font size="3" face="arial" color="66CC00">
    Total do Pomar = <?php echo $total."\n";?>
    <BR>
    <BR>
    </font>
    </div>
    <FORM><INPUT TYPE="button" VALUE="Voltar" onClick="history.go(-
    1);return true;"></FORM>
    </body>
    </html>
```

Anexo 13 – Gráfico Limpezas/Data

```
<html><head>
<title>Limpeza</title>
<script type="text/javascript" src="jscharts.js"></script>
</head>
<body>
<?php
    $array = array();

    $conexao =
mysql_connect('mysql2.000webhost.com','a3479280_root','ulht2012');
    mysql_select_db('a3479280_pomar',$conexao);

    $sql="select Data_Intervencao, Quantidade from intervencoes
where Motivo = 'Limpeza' ";
    $resultado = mysql_query($sql) or die ("Erro
.: ".mysql_error());

    while($row=mysql_fetch_assoc($resultado))
        $array[]=$row;
    $total=0;
    foreach ($array as $a){

        $total = $total + $a['Quantidade'];
        $arrayi[] =
"['".$a['Data_Intervencao'].",'".$a['Quantidade']."]";

    }
    if (mysql_num_rows($resultado)>0)
        $s = implode (',', $arrayi);
        else {
            $erro='Sem Dados';
        }

?>
<font size="3" face="arial" color="66CC00">
<?php echo $erro."\n";?>
<BR>
<BR>
</font>

<div id="graph">Loading graph...</div>
<script type="text/javascript">

    var myData = new Array(<?php echo $s ?>);
    var myChart = new JSChart('graph', 'bar');
    myChart.setDataArray(myData);
    myChart.setBarColor('#42aBdB');
    myChart.setBarOpacity(0.8);
    myChart.setBarBorderColor('#D9EDF7');
    myChart.setBarValues(false);
    myChart.setTitleColor('#8C8383');
    myChart.setAxisColor('#777E81');
    myChart.setAxisNameX('Data Intervençoes');
    myChart.setAxisNameY('');
    myChart.setTitle('Limpeza por Pore');
    myChart.setAxisValuesColorX('#777E81');
    myChart.setAxisValuesColorY('#FFFFFF');
```

```
        myChart.setSize(500, 350);
        myChart.draw();
    </script>
</div>
<font size="3" face="arial" color="66CC00">
Total de Limpezas = <?php echo $total."\n";?>
<BR>
<BR>
</font>
<FORM><INPUT TYPE="button" VALUE="Voltar" onClick="history.go(-
1);return true;"></FORM>
</body>
</html>
```

Anexo 14 – Gráfico Tratamentos/Data

```
<html><head>
<title>Tratamento</title>
<script type="text/javascript" src="jscharts.js"></script>
</head>
<body>
<?php
    $array = array();

    $conexao =
mysql_connect('mysql2.000webhost.com','a3479280_root','ulht2012');
    mysql_select_db('a3479280_pomar',$conexao);

    $sql="select Data_Intervencao, Quantidade from intervencoes
where Motivo = 'Tratamento' ";
    $resultado = mysql_query($sql) or die ("Erro
.: ".mysql_error());

    while($row=mysql_fetch_assoc($resultado))
    $array[]=$row;
    $total=0;

    foreach ($array as $a){

        $total = $total + $a['Quantidade'];
        $arrayi[] =
"['".$a['Data_Intervencao']. "','.$a['Quantidade']. '"]";
    }
    if (mysql_num_rows($resultado)>0)
    $s = implode (',', $arrayi);
    else {
        $erro='Sem Dados';
    }

?>
<font size="3" face="arial" color="66CC00">
<?php echo $erro."\n";?>
<BR>
<BR>
</font>

<div id="graph">Loading graph...</div>
<script type="text/javascript">

    var myData = new Array(<?php echo $s ?>);
    var myChart = new JSChart('graph', 'bar');
    myChart.setDataArray(myData);
    myChart.setBarColor('#42aBdB');
    myChart.setBarOpacity(0.8);
    myChart.setBarBorderColor('#D9EDF7');
    myChart.setBarValues(false);
    myChart.setTitleColor('#8C8383');
    myChart.setAxisColor('#777E81');
    myChart.setAxisNameX('Data Intervençoes');
    myChart.setAxisNameY('');
    myChart.setTitle('Tratamento por Core');
    myChart.setAxisValuesColorX('#777E81');
    myChart.setAxisValuesColorY('FFFFFF');
```

```
        myChart.setSize(500, 350);
        myChart.draw();
    </script>
</div>
<font size="3" face="arial" color="66CC00">
Total de tratamientos= <?php echo $total."\n";?>
<BR>
<BR>
</font>
<FORM><INPUT TYPE="button" VALUE="Voltar" onClick="history.go(-
1);return true;"></FORM>
</body>
</html>
```


Anexo 15 – Outros/Data

```
<html><head>
<title>Outros Motivos</title>
<script type="text/javascript" src="jscharts.js"></script>
</head>
<body>
<?php
    $array = array();

    $conexao =
mysql_connect('mysql2.000webhost.com','a3479280_root','ulht2012');
    mysql_select_db('a3479280_pomar',$conexao);

    $sql="select Data_Intervencao, Quantidade from intervencoes
where Motivo = 'Outro' ";
    $resultado = mysql_query($sql) or die ("Erro
.: ".mysql_error());

    while($row=mysql_fetch_assoc($resultado))
        $array[]=$row;
    $total=0;
    foreach ($array as $a){

        $total = $total + $a['Quantidade'];
        $arrayi[] =
"['".$a['Data_Intervencao'].",'".$a['Quantidade']."]";

    }
    if (mysql_num_rows($resultado)>0)
        $s = implode (',',$arrayi);
    else {
        $erro='Sem Dados';
    }

?>
<font size="3" face="arial" color="66CC00">
<?php echo $erro."\n";?>
<BR>
<BR>
</font>

<div id="graph"></div>
<script type="text/javascript">

    var myData = new Array(<?php echo $s ?>);
    var myChart = new JSChart('graph', 'bar');
    myChart.setDataArray(myData);
    myChart.setBarColor('#42aBdB');
    myChart.setBarOpacity(0.8);
    myChart.setBarBorderColor('#D9EDF7');
    myChart.setBarValues(false);
    myChart.setTitleColor('#8C8383');
    myChart.setAxisColor('#777E81');
    myChart.setAxisNameX('Data Intervencao');
    myChart.setAxisNameY('');
    myChart.setTitle('Outros Motivos por Core');
    myChart.setAxisValuesColorX('#777E81');
    myChart.setAxisValuesColorY('FFFFFF');
```

```
        myChart.setSize(500, 350);
        myChart.draw();
    </script>
</div>
<font size="3" face="arial" color="66CC00">
Total de outros motivos= <?php echo $total."\n";?>
<BR>
<BR>
</font>
<FORM><INPUT TYPE="button" VALUE="Voltar" onClick="history.go(-
1);return true;"></FORM>
</body>
</html>
```

Anexo 16 – Logout

```
<?php
if( !session_id() ) session_start();
//termina vari'el de sessão/span>
unset($_SESSION['nome']);
?>
<html>
<body>
<span style="padding-left:60px">

<BR>
<BR>
<BR>
<BR>
<span style="padding-left:210px">
<!--reencaminha para a homepage-->
<a href="default.php">Fazer login novamente</a>
</body>
</html>
```

Source Code - Aplicação Java para Android

Anexo 17 – login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal" >

    <TableLayout
        android:id="@+id/tableLayout1"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:background="@drawable/img_3261">

        <TableRow
            android:id="@+id/tableRow1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" >

            <TextView
                android:id="@+id/textView1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceLarge" />

        </TableRow>

        <TableRow
            android:id="@+id/tableRow2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" >

            <TextView
                android:id="@+id/textView2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceLarge" />

        </TableRow>

        <TableRow
            android:id="@+id/tableRow3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" >

        </TableRow>

        <TableRow
            android:id="@+id/tableRow4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" >

        </TableRow>

        <Button
            android:id="@+id/btLogin"
            android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:text="Login" android:autoLink="web"/>

<Button
    android:id="@+id/btWeb"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="AppWeb" />

    <Button
        android:layout_width="wrap_content"
        android:text="Ler Árvore"
        android:id="@+id/btArvoreMain"
        android:layout_height="wrap_content"/>
    <Button android:text="Sincronizar Dados"
android:id="@+id/btEnvia" android:layout_width="wrap_content"
android:layout_height="wrap_content"></Button>
    <Button android:text="Consultar" android:id="@+id/btConsulta"
android:layout_width="wrap_content"
android:layout_height="wrap_content"></Button>

<Button
    android:id="@+id/btsair"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Sair" />

</TableLayout>

</LinearLayout>

```

Anexo 18 – main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal" android:background="@drawable/img_3261">

    <TableLayout
        android:id="@+id/TableLayout1"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:background="@color/background"
        android:orientation="horizontal" >

        <TextView
            android:id="@+id/tvuser"
            android:layout_gravity="left"
            android:text="User Name:"
            android:textAppearance="?android:attr/textAppearanceMedium" />

        <EditText
            android:id="@+id/etuser"
            android:layout_width="200dp"
            android:inputType="textPersonName" />

        <TextView
            android:id="@+id/tvpass"
            android:layout_gravity="left"
            android:text="Password:"
            android:textAppearance="?android:attr/textAppearanceMedium" />

        <EditText
            android:id="@+id/etpass"
            android:layout_width="203dp"
            android:inputType="textPassword" />

        <Button
            android:id="@+id/btok"
            android:layout_width="97dp"
            android:text="OK" />

        <Button
            android:id="@+id/btsairlogin"
            android:layout_width="86dp"
            android:text="Voltar" />
    </TableLayout>
</LinearLayout>
```

Anexo 19 - arvregint.xml

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    android:background="@drawable/img_3261">
    <TabHost android:id="@+id/tabhost" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
        <LinearLayout android:id="@+id/linearLayout1"
        android:layout_width="fill_parent" android:layout_height="fill_parent"
        android:orientation="vertical">
            <TabWidget android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:id="@android:id/tabs"></TabWidget>
            <FrameLayout android:layout_width="fill_parent"
            android:layout_height="fill_parent" android:id="@android:id/tabcontent">
                <LinearLayout android:layout_width="fill_parent"
                android:layout_height="fill_parent" android:id="@+id/tab1" >

                    <ScrollView
                        android:layout_width="fill_parent"
                        android:layout_height="fill_parent"
                        android:background="@color/background"
                        >

                        <TableLayout
                            android:id="@+id/tableLayout1"
                            android:layout_width="fill_parent"
                            android:layout_height="fill_parent" >

                                <TextView
                                    android:id="@+id/textView1"

                                    android:text="ID Árvore:"

                                android:textAppearance="?android:attr/textAppearanceMedium" />

                                <EditText
                                    android:id="@+id/IDArvore"
                                    android:layout_width="198dp"
                                    android:editable="false">

                                    <requestFocus />
                                </EditText>

                                <Button
                                    android:id="@+id/btarvore"
                                    android:layout_width="wrap_content"
                                    android:layout_height="wrap_content"
                                    android:text="Ler Árvore" />

                                <TextView
                                    android:id="@+id/textView7"

                                    android:text="ID Produtor"
```



```

android:textAppearance="?android:attr/textAppearanceMedium" />

        <EditText
            android:id="@+id/IDProdutor"
            android:layout_width="198dp"
            android:editable="false"/>

        <TextView
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceMedium"
            android:layout_width="wrap_content" android:id="@+id/textView2"
            android:text="Data Plantação:"></TextView>
            <DatePicker
                android:layout_gravity="left" android:id="@+id/datePicker1"
                android:layout_height="fill_parent"
                android:layout_width="fill_parent"></DatePicker>
                <TextView
                    android:id="@+id/textView8"
                    android:text="Data QRCode"

android:textAppearance="?android:attr/textAppearanceMedium" />
            <DatePicker
                android:id="@+id/datePicker2" android:layout_width="fill_parent"
                android:layout_height="fill_parent"></DatePicker>

            <TextView
                android:id="@+id/textView4"

                android:text="Tipo de árvore:"

android:textAppearance="?android:attr/textAppearanceMedium" />

        <Spinner
            android:id="@+id/spinnertipo"
            android:layout_width="185dp"
            android:layout_height="wrap_content"
            android:prompt="@array/ListaArvores" />

        <TextView
            android:id="@+id/textView6"

android:textAppearance="?android:attr/textAppearanceMedium" />

        <Button
            android:id="@+id/btleitura"

            android:text="Limpar" />

        <Button
            android:id="@+id/btguardar"

            android:text="Guardar" />

        <Button

```

```

        android:id="@+id/btvoltarmenu"

        android:layout_gravity="right"

        android:text="Voltar" />

    </TableLayout>
</ScrollView>

</LinearLayout>

<LinearLayout android:layout_width="fill_parent"
android:layout_height="fill_parent" android:id="@+id/tab2">
    <ScrollView

        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:background="@color/background">

        <TableLayout
            android:id="@+id/tableLayout2"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent" >

            <TextView android:text=""
android:layout_height="wrap_content" android:layout_width="fill_parent"
android:id="@+id/intervencao/txtidProdutor"
android:textAppearance="?android:attr/textAppearanceMedium"
android:visibility="gone"></TextView>
            <TextView android:text="ID Árvore:"
android:layout_height="wrap_content" android:layout_width="fill_parent"
android:id="@+id/intervencao/txtid"
android:textAppearance="?android:attr/textAppearanceMedium"></TextView>
            <Button android:text="Ler Árvore"
android:id="@+id/intervencao/btler" android:layout_height="wrap_content"
android:layout_width="fill_parent"></Button>
            <EditText android:id="@+id/intervencao/edid"
android:text="" android:layout_height="wrap_content"
android:layout_width="fill_parent" android:editable="false"></EditText>
            <TextView android:text="Data de Intervenção:"
android:layout_height="wrap_content" android:layout_width="fill_parent"
android:id="@+id/intervencao/data"
android:textAppearance="?android:attr/textAppearanceMedium"></TextView>
            <DatePicker android:id="@+id/intervencao/dataIntervencao"
android:layout_width="wrap_content"
android:layout_height="wrap_content"></DatePicker>
            <TextView android:text="Escolha o Motivo:"
android:layout_height="wrap_content" android:layout_width="fill_parent"
android:id="@+id/intervencao/textView1"
android:textAppearance="?android:attr/textAppearanceMedium"></TextView>
            <Spinner android:id="@+id/intervencao/motivo"
android:layout_width="fill_parent" android:layout_height="wrap_content"
android:prompt="@array/ListaMotivos"></Spinner>
            <TextView android:text="Quantidade:"
android:id="@+id/intervencao/txtQuantidade" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:visibility="invisible"
android:textAppearance="?android:attr/textAppearanceMedium"></TextView>
            <EditText android:text=""
android:id="@+id/intervencao/edQuant" android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:visibility="invisible"></EditText>

```

```

        <TextView android:text="Observação:"
android:id="@+intervencao/txtObs" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:visibility="invisible"
android:textAppearance="?android:attr/textAppearanceMedium"></TextView>
        <EditText android:id="@+intervencao/edObs"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:inputType="textMultiLine" android:visibility="invisible" />

        <Button android:text="Gravar"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:id="@+intervencao/btgravar"></Button>
        <Button android:text="Limpar"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:id="@+intervencao/btproximo"></Button>
        <Button android:text="Voltar"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:id="@+intervencao/btvoltarmenu"></Button>
    </TableLayout>
</ScrollView>
</LinearLayout>

</FrameLayout>
</LinearLayout>
</TabHost>
</LinearLayout>

```

Anexo 20 – consulta_arvore.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:background="@drawable/img_3261" android:layout_height="fill_parent"
    android:id="@+id/LinearLayout1" android:layout_width="fill_parent"
    android:orientation="vertical">
    <LinearLayout android:id="@+id/LinearLayout3"
        android:orientation="horizontal" android:alwaysDrawnWithCache="true"
        android:layout_height="wrap_content" android:layout_width="fill_parent"
        android:background="@color/background">
        <TextView android:text="Filtrar Consulta:"
            android:textColor="#ffff00" android:id="@+id/textView1"
            android:layout_height="wrap_content" android:textSize="20sp"
            android:layout_width="wrap_content"></TextView>
        <Spinner android:id="@+id/spinnerConsulta"
            android:prompt="@array/ListaFiltros" android:layout_height="wrap_content"
            android:layout_width="wrap_content"></Spinner>
    </LinearLayout>

    <ScrollView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">

        <LinearLayout android:id="@+id/LinearLayoutmain"
            android:orientation="vertical" android:visibility="visible"
            android:layout_height="fill_parent" android:layout_width="fill_parent">

            <TableLayout android:id="@+id/tableLayout1"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
            android:layout_gravity="center" android:background="@color/background">

                <LinearLayout android:id="@+id/LinearLayoutmain2"
                    android:orientation="vertical" android:visibility="visible"
                    android:layout_height="fill_parent" android:layout_width="fill_parent">
                    <TextView android:text="Consultar árvores"
                        android:id="@+id/txtTitulo" android:textColor="#ffff00"
                        android:layout_height="wrap_content" android:textSize="25sp"
                        android:layout_width="fill_parent"></TextView>
                    <Button android:layout_height="wrap_content"
                        android:layout_width="fill_parent" android:id="@+id/LerArvId"
                        android:text="Ler Árvore"></Button>
                    <LinearLayout android:layout_height="wrap_content"
                        android:id="@+id/LinearLayouttxtId" android:orientation="horizontal"
                        android:layout_width="wrap_content">
                        <EditText android:text="" android:id="@+id/edidArvId"
                            android:visibility="visible" android:layout_height="wrap_content"
                            android:layout_width="100dip" android:editable="false"></EditText>
                        <ImageButton android:id="@+id/btPesquisaId"
                            android:background="@drawable/lupa_pesquisa" android:layout_width="40dip"
                            android:layout_height="40dip"></ImageButton>
                    </LinearLayout>
                    <TextView android:text="TextView" android:id="@+id/txtid"
                        android:layout_height="wrap_content" android:textSize="20sp"
                        android:layout_width="fill_parent"></TextView>
                </TableLayout>
            </LinearLayout>
        </ScrollView>
    </LinearLayout>
```

```

        <TextView android:text="TextView" android:id="@+id/txtP"
android:layout_width="fill_parent" android:layout_height="wrap_content"
android:textSize="20sp"></TextView>
        <TextView android:text="TextView" android:id="@+id/txtQR"
android:layout_width="fill_parent" android:layout_height="wrap_content"
android:textSize="20sp"></TextView>
        <TextView android:text="TextView" android:id="@+id/txtTipo"
android:layout_width="fill_parent" android:layout_height="wrap_content"
android:textSize="20sp"></TextView>
        <TextView android:text="TextView"
android:id="@+id/txtestado" android:layout_width="fill_parent"
android:layout_height="wrap_content" android:textSize="20sp"
android:visibility="gone"></TextView>
        <TextView android:text="TextView"
android:id="@+id/txtidProdutor" android:layout_width="fill_parent"
android:layout_height="wrap_content" android:textSize="20sp"
android:visibility="gone"></TextView>

        <LinearLayout android:layout_height="wrap_content"
android:id="@+id/linearLayout2" android:layout_width="wrap_content"
android:orientation="horizontal">
            <Button android:layout_height="wrap_content"
android:text="Anterior" android:id="@+id/btRegAnt"
android:layout_width="wrap_content"></Button>
            <Button android:text="Próximo" android:id="@+id/btProxReg"
android:layout_width="wrap_content"
android:layout_height="wrap_content"></Button>
            <Button android:text="Apagar"
android:id="@+id/btapagarReg" android:layout_width="wrap_content"
android:layout_height="wrap_content"></Button>
        </LinearLayout>
        </LinearLayout>
        <Button android:text="Menu Principal" android:id="@+id/btVoltar"
android:layout_width="wrap_content"
android:layout_height="wrap_content"></Button>
    </TableLayout>

</LinearLayout>
</ScrollView>
<ScrollView android:layout_width="fill_parent"
android:layout_height="wrap_content">
    <LinearLayout android:id="@+id/linearLayoutInter"
android:layout_width="fill_parent" android:layout_height="fill_parent"
android:orientation="vertical" android:alwaysDrawnWithCache="true"
android:visibility="gone">

        <TableLayout android:id="@+id/tableLayout1"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
android:background="@color/background">
            <LinearLayout android:id="@+id/linearLayoutmain3"
android:orientation="vertical" android:visibility="visible"
android:layout_height="fill_parent" android:layout_width="fill_parent">

                <TextView android:textSize="25sp" android:text="Filtrar por intervenção"
android:id="@+id/textView1" android:textColor="#fff0ff"
android:layout_height="wrap_content"
android:layout_width="wrap_content"></TextView>
            
```

```

        <Button android:layout_height="wrap_content"
android:layout_width="fill_parent" android:id="@+consultainter/LerArvore"
android:text="Ler Árvore"></Button>
        <LinearLayout android:layout_height="wrap_content"
android:id="@+id/LinearLayoutxpto" android:orientation="horizontal"
android:layout_width="wrap_content">
            <EditText android:text="" android:id="@+consultainter/edidInter"
android:visibility="visible" android:layout_height="wrap_content"
android:layout_width="100dip" android:editable="false"></EditText>
            <ImageButton android:id="@+consultainter/btPesquisa"
android:background="@drawable/lupa_pesquisa" android:layout_width="40dip"
android:layout_height="40dip"></ImageButton>
        </LinearLayout>
        <TextView android:text="" android:id="@+consultainter/txtidArv"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:textSize="20sp"></TextView>
        <TextView android:text="" android:id="@+consultainter/txtIdInter"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:textSize="20sp"></TextView>
        <TextView android:text="" android:id="@+consultainter/txtData"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:textSize="20sp"></TextView>
        <TextView android:text="" android:id="@+consultainter/txtMotivo"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:textSize="20sp"></TextView>
        <TextView android:text=""
android:id="@+consultainter/txtQuantidade"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:textSize="20sp"></TextView>
        <TextView android:text=""
android:id="@+consultainter/txtEstadoInterCons"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:textSize="20sp" android:visibility="gone"></TextView>
        <TextView android:text=""
android:id="@+consultainter/txtIdProdConsInter"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:textSize="20sp" android:visibility="gone"></TextView>
        <TextView android:text="" android:id="@+consultainter/txtObs"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:textSize="20sp"></TextView>
        <LinearLayout android:layout_height="wrap_content"
android:id="@+id/LinearLayout2" android:layout_width="wrap_content"
android:orientation="horizontal">
            <Button android:layout_height="wrap_content" android:text="Anterior"
android:id="@+consultainter/btRegAnt"
android:layout_width="wrap_content"></Button>
            <Button android:text="Próximo" android:id="@+consultainter/btProxReg"
android:layout_width="wrap_content"
android:layout_height="wrap_content"></Button>
            <Button android:text="Apagar" android:id="@+consultainter/btApagarReg"
android:layout_width="wrap_content"
android:layout_height="wrap_content"></Button>
        </LinearLayout>
        </LinearLayout>
        <Button android:text="Menu Principal"
android:id="@+consultainter/btVoltar" android:layout_width="wrap_content"
android:layout_height="wrap_content"></Button>
    </TableLayout>
</LinearLayout>

```



```

</ScrollView>
<ScrollView android:layout_width="fill_parent"
android:layout_height="wrap_content">
<LinearLayout android:id="@+id/linearLayoutInterMotivo"
android:layout_width="fill_parent" android:layout_height="fill_parent"
android:orientation="vertical" android:alwaysDrawnWithCache="true"
android:visibility="gone">
<TableLayout android:id="@+id/tableLayout1"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
android:background="@color/background">
    <LinearLayout android:id="@+id/linearLayoutmain4"
android:orientation="vertical" android:visibility="visible"
android:layout_height="fill_parent" android:layout_width="fill_parent"
android:background="@color/background">
        <TextView android:textSize="25sp" android:text="Filtrar por
Motivo" android:id="@+id/textViewFiltrar" android:textColor="#fff0ff"
android:layout_height="wrap_content"
android:layout_width="wrap_content"></TextView>
        <Button android:layout_height="wrap_content"
android:layout_width="fill_parent"
android:id="@+id/consultaintermotivo/LerArvore" android:text="Ler
Arvore"></Button>
        <LinearLayout android:layout_height="wrap_content"
android:id="@+id/linearLayoutxptoMotivo" android:orientation="horizontal"
android:layout_width="wrap_content">
            <Spinner android:id="@+id/spinnerConsultaMotivo"
android:prompt="@array/ListaMotivos" android:layout_height="wrap_content"
android:layout_width="wrap_content"></Spinner>
            <EditText android:text=""
android:id="@+id/consultaintermotivo/edidInter" android:visibility="visible"
android:layout_height="wrap_content" android:layout_width="100dip"
android:editable="false"></EditText>
            <ImageButton android:id="@+id/consultaintermotivo/btPesquisa"
android:background="@drawable/lupa_pesquisa" android:layout_width="40dip"
android:layout_height="40dip"></ImageButton>
        </LinearLayout>
        <TextView android:text=""
android:id="@+id/consultaintermotivo/txtidArv"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:textSize="20sp"></TextView>
        <TextView android:text=""
android:id="@+id/consultaintermotivo/txtIdInter"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:textSize="20sp"></TextView>
        <TextView android:text=""
android:id="@+id/consultaintermotivo/txtData"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:textSize="20sp"></TextView>
        <TextView android:text=""
android:id="@+id/consultaintermotivo/txtMotivo"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:textSize="20sp"></TextView>
        <TextView android:text=""
android:id="@+id/consultaintermotivo/txtQuantidade"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:textSize="20sp"></TextView>
        <TextView android:text=""
android:id="@+id/consultaintermotivo/txtEstadoInterCons"

```

```

android:layout_width="wrap_content" android:layout_height="wrap_content"
android:textSize="20sp" android:visibility="gone"></TextView>
    <TextView android:text=""
android:id="@+consultaintermotivo/txtIdProdConsInter"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:textSize="20sp" android:visibility="gone"></TextView>
    <TextView android:text=""
android:id="@+consultaintermotivo/txtObs" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:textSize="20sp"></TextView>
    <LinearLayout android:layout_height="wrap_content"
android:id="@+id/LinearLayout2" android:layout_width="wrap_content"
android:orientation="horizontal">
        <Button android:layout_height="wrap_content" android:text="Anterior"
android:id="@+consultaintermotivo/btRegAnt"
android:layout_width="wrap_content"></Button>
        <Button android:text="Próximo"
android:id="@+consultaintermotivo/btProxReg"
android:layout_width="wrap_content"
android:layout_height="wrap_content"></Button>
        <Button android:text="Apagar"
android:id="@+consultaintermotivo/btApagarReg"
android:layout_width="wrap_content"
android:layout_height="wrap_content"></Button>
    </LinearLayout>
</LinearLayout>
    <Button android:text="Menu Principal"
android:id="@+consultaintermotivo/btVoltar"
android:layout_width="wrap_content"
android:layout_height="wrap_content"></Button>
</TableLayout>
</LinearLayout>
</ScrollView>
</LinearLayout>

```


Anexo 21 – filtaintervencoes.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/LinearLayout1" android:layout_width="fill_parent"
    android:layout_height="fill_parent" android:background="@drawable/img_3261"
    android:orientation="vertical">
    <TextView android:textSize="26sp" android:text="Filtrar por
    intervenção" android:id="@+id/textView1" android:textColor="#fff0ff"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"></TextView>
    <Button android:layout_height="wrap_content"
    android:layout_width="fill_parent" android:id="@+id/consultainter/LerArvore"
    android:text="Ler Árvore"></Button>
    <TextView android:text="" android:id="@+id/consultainter/txtidArv"
    android:layout_width="wrap_content" android:layout_height="wrap_content"
    android:textSize="20sp"></TextView>
    <TextView android:text="" android:id="@+id/consultainter/txtIdInter"
    android:layout_width="wrap_content" android:layout_height="wrap_content"
    android:textSize="20sp"></TextView>
    <TextView android:text="" android:id="@+id/consultainter/txtData"
    android:layout_width="wrap_content" android:layout_height="wrap_content"
    android:textSize="20sp"></TextView>
    <TextView android:text="" android:id="@+id/consultainter/txtMotivo"
    android:layout_width="wrap_content" android:layout_height="wrap_content"
    android:textSize="20sp"></TextView>
    <TextView android:text=""
    android:id="@+id/consultainter/txtQuantidade"
    android:layout_width="wrap_content" android:layout_height="wrap_content"
    android:textSize="20sp"></TextView>
    <LinearLayout android:layout_height="wrap_content"
    android:id="@+id/LinearLayout2" android:layout_width="wrap_content"
    android:orientation="horizontal">
    <Button android:layout_height="wrap_content" android:text="Anterior"
    android:id="@+id/consultainter/btRegAnt"
    android:layout_width="wrap_content"></Button>
    <Button android:text="Próximo" android:id="@+id/consultainter/btProxReg"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"></Button>
    <Button android:text="Apagar" android:id="@+id/consultainter/btApagarReg"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"></Button>
    </LinearLayout>
    <Button android:text="Menu Principal"
    android:id="@+id/consultainter/btVoltar" android:layout_width="wrap_content"
    android:layout_height="wrap_content"></Button>

</LinearLayout>
```

Anexo 22 – AppPomar.java

```
package pt.com.android;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.BufferedInputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLConnection;
import java.net.URLEncoder;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.entity.ByteArrayEntity;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.params.BasicHttpParams;
import org.apache.http.params.HttpConnectionParams;
import org.apache.http.params.HttpParams;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.AlertDialog.Builder;
import android.app.ProgressDialog;
import android.app.TabActivity;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.net.NetworkInfo.State;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.view.ViewStub;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
```

```

import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.LinearLayout;
import android.widget.Spinner;
import android.widget.TabHost;
import android.widget.TabHost.TabSpec;
import android.widget.TextView;
import com.google.zxing.*;

public class AppPomar extends Activity {
    /** Called when the activity is first created. */
    final Context context = this;
    enum ViewMode { MAIN, CONSULTAS, LERARVORES};
    private ViewMode mViewMode;
    EditText etuser, etpass, IDProdutor, IDArvore, idArvoreIntervencao,
    DataPlantacao, DataQRCode, DataColheita,
    Quantidade, edQuantInter, edObs;
    TextView
    error, txtid, txtCol, txtP, txtQR, txtTipo, txtQuantidade, txtObservacao, txtI
    dProdInter, txtidArvInter, txtDataInter, txtMotivoInter, txtDesc, txtQuantI
    nter, txtestado, txtidProdutor;
    Button
    btok, btEnvia, btsairlogin, btlogin, btsair, btarvore, btvoltarmenu, btWeb, bt
    guardar, btArvoreMain, btConsulta, btLimparArv, btLimparInter;
    Button btGravaInter, btVoltarInter, btLerInter, btProximoInter;
    //Botda pçna intervenções
    ImageButton ImgBtPesquisa, ImgBtPesquisaMotivo, ImgBtIdPesquisa;
    DatePicker dataP, dataQ, dataCol, dataIntervencao;
    String id, idArvore, idArvoreInter, pass, idArvoreCons="";
    Spinner spinner_Consulta;
    Spinner tipoarvore, spinner_motivo;
    String qrdata, intervencao, plantacao, motivo;
    //OBJECTOS da Consulta
    Button
    btLerArvIdCons, btRegAnt, btProxReg, btMenuP, btMenuPConsInter, btRegAntInt
    er, btProxRegInter, btApagarRegInter, btRegAntData, btProxRegData, btLerCon
    sInter;
    Button
    btRegAntMotivo, btProxRegMotivo, btApagarMotivo, btLerArvMotivo, btMPmotiv
    o, btApagarArv;
    Spinner spMotivoCons;
    EditText edIdInter, edIdMotivo, edIdArv;
    TextView
    txtObsInter, txtArvConsInter, txtIdConsInter, txtDataInterCons, txtMotivoI
    nterCons, txtQuantInterCons, txtEstadoConsInter, txtIdProdConsInter;
    TextView
    txtObsMotivo, txtArvConsMotivo, txtIdConsInterMotivo, txtDataInterMotivo,
    txtQuantConsMotivo, txtEstadoConsInterMotivo, txtIdProdConsMotivo, txtMot
    ivoInterConsMotivo;
    String tipo;
    //////////////////////////////////////
    SQLiteDatabase BD =null;
    Cursor cursor, cursor_consulta;
    LinearLayout
    mainLayout, intervencaoLayout, motivoLayout, dataLayout; //Layouts para
    consultas

```

```

        int
        quantidade,id_arvore,data_plantacao,data_scanner,tipo_arvore,id_Produto,
        estado_CA,motivoInter,idInterCons,id_arvoreCons,estadoConsInter,idP
        roductorConsInter,obsCons;
        public static final int TIMEOUT_MILLISEC = 30 * 1000;
        boolean
        login=false,tab1,tab2,consultaTodas,consultaData=false,consultaMotivos
        =false,ConsultaInter=false;
        public int erroLogin=0;
        @Override
        public void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            main();
        }

        private class LoginServerTask extends AsyncTask<String, Void, Void>
        {
            ProgressDialog dialog;
            boolean erro=true,erroServer=false,erro_Login=true;
            @Override
            protected void onPreExecute() //mostra o dialog enquanto estiver
            a executar a tarefa
            {
                dialog = new ProgressDialog (AppPomar.this);
                this.dialog.setMessage("A verificar dados...");
                this.dialog.show();
                this.dialog.setCancelable(false);
            }

            // acaba automaticamente
            @Override
            protected Void doInBackground(String... args)//estes args s
            urls das paginas php
            {

                ArrayList<BasicNameValuePair> postParameters = new
                ArrayList<BasicNameValuePair>();
                postParameters.add(new BasicNameValuePair("user",
                etuser.getText().toString()));
                postParameters.add(new BasicNameValuePair("pass",
                etpass.getText().toString()));
                String response = null;
                String resposta =null;
                try {
                    if(serverIsOnline())//funcao que verifica se o servidor
                    est
                    inline
                    {
                        response = CustomHttpClient.executeHttpPost(args[0],
                        postParameters);
                        resposta = response.toString();
                        resposta = resposta.substring(0,1);
                        Log.i("VERIFICA RESPOSTA",response.toString()+"
                        resposta:"+resposta.toString());
                        if(resposta.equals("1")) //faz a verificac
                        dados no servidor atraves da resposta vinda da p
                        {
                            InserirUser(); //insere na bd interna
                            id = etuser.getText().toString();
                            pass=etpass.getText().toString();
                            recebeDados(args[1]); //recebe os dados referentes
                            cores

```

```

        recebeIntervencoes(args[2]); //recebe os dados
referentes   interven es nas  ores
        erro_Login=false;
    }

    else
    {
        if(erroLogin<=3)
        {

            erroLogin++;
            Log.i("ERROLOGIN",""+erroLogin);
            erro_Login=true;
        }
        else
        {
            erro=true;
            Log.i("depois do erro=true",""+erroLogin);
        }
    }
    else
    {
        erroServer=true;
    }
}

    catch (Exception e) {

    }

    return null;
}
@Override
protected void onPostExecute(Void unused) //esta fun  o   executada
depois da fun  o principal.
{
    Log.i("ERRO_LOGIN",""+erroLogin);
    if(erroServer==true)
    {
        msg("Aviso!", "Servidor encontra-se em modo offline.
Tente mais tarde.");
        dialog.dismiss();
    }
    else
    {
        if(erro_Login==false)
        {
            dialog.dismiss();
            login=true;
            main();
            msg("Conclu", "Login Correto. Dados sincronizados!");
        }
        else if((erro_Login==true) && (erroLogin<=3))
        {
            msg("Erro", "Login incorrecto!");
            dialog.dismiss();
        }
        else
        {

```

```

        msg("Erro", "Efectue o registo na app web, clicando no
botão 'AppWeb'");
        dialog.dismiss();
        erro_Login=true;
        erro=true;
    }
}

}

public void LoginServer() {
    if(isOnline())//Funcao que verifica se o dispositivo tem
internet activa
    {
        LoginServerTask task = new LoginServerTask(); //Cria a tarefa
        //task.execute(new String[] {
"http://192.168.2.101/login.php", "http://192.168.2.101/getArvores.php"
, "http://192.168.2.101/getIntervencoes.php" });
        task.execute(new String[] {
"http://pomardigital.net76.net/login.php", "http://pomardigital.net76.n
et/getArvores.php", "http://pomardigital.net76.net/getIntervencoes.php"
});
    }
    else
    {
        msg("Aviso!", "Necessita de conexão internet para verificar se
existe no servidor. Caso contrário faça o registo da app na aplicação.");
    }
}

private class SincronizaServerTask extends
AsyncTask<String,Void,Void>
{
    ProgressDialog dialog;
    boolean erro=false;
    @Override
    protected void onPreExecute() {
        dialog = new ProgressDialog (AppPomar.this);
        this.dialog.setMessage("A sincronizar dados...");
        this.dialog.show();
        this.dialog.setCancelable(false);
    }

    // acaba automaticamente
    @Override
    protected Void doInBackground(String... args) {
        if(serverIsOnline())//funcao que verifica se o servidor
está online
        {
            EnviaDados(args[0]);
            EnviaDadosIntervencao(args[1]);
            recebeIntervencoesApagadas(args[2]);
            recebeDadosApagados(args[3]);
            recebeDadosActualizados(args[4]);
            recebeIntervencoesActualizadas(args[5]);

        }
        else
        {
            erro=true;
        }
    }
}

```

```

        return null;
    }
    @Override
    protected void onPostExecute(Void unused) {
        if (erro == false)
        {
            dialog.dismiss();
            login = true;
            main();
            msg("Conclu", "Dados sincronizados!");
        }
        else
        {
            msg("Aviso!", "Servidor encontra-se em modo offline.
Tente mais tarde.");
            dialog.dismiss();
        }
    }
}

public void SincronizaServer()
{
    if (isOnline()) // Funcao que verifica se o dispositivo tem
internet activa
    {
        SincronizaServerTask task = new SincronizaServerTask();
        //task.execute(new String[] {
"http://192.168.2.101/RegistaArvore.php", "http://192.168.2.101/Regista
Intervencoes.php"});
        //task.execute(new String[] {
"http://172.19.100.140/RegistaArvore.php", "http://172.19.100.140/Regis
taIntervencoes.php"});
        task.execute(new String[]
{"http://pomardigital.net76.net/RegistaArvore.php", "http://pomardigita
l.net76.net/RegistaIntervencoes.php", "http://pomardigital.net76.net/ge
tIntervencoesInativas.php", "http://pomardigital.net76.net/getArvoresIn
ativas.php", "http://pomardigital.net76.net/getArvoresEstado.php", "http
://pomardigital.net76.net/getIntervencoesEstado.php"});
    }
    else
    {
        msg("Aviso!", "Necessita de conex棒*internet para verificar se
existe no servidor. Caso contr□o fa 破o registo da □ore na aplica 磯.");
    }
}

public void main () {
    setContentView(R.layout.main);
    mViewMode = ViewMode.MAIN;
    inicializaobjectos();
    listener();
    OpenCreateBD();
    if (login == false)
    {
        btArvoreMain.setVisibility(Button.GONE);
        btConsulta.setVisibility(Button.GONE);
        btEnvia.setVisibility(Button.GONE);
    }
}

```

```

    }

    public boolean isOnline() //Função para verificar conectividade
    {
        ConnectivityManager cm =
            (ConnectivityManager)
            getSystemService(Context.CONNECTIVITY_SERVICE);

        return cm.getActiveNetworkInfo() != null &&
            cm.getActiveNetworkInfo().isConnectedOrConnecting();
    }

    public boolean serverIsOnline(){
    try {
        //URL url = new URL("http://192.168.2.101");
        //URL url = new URL("http://172.19.100.140");
        URL url = new URL("http://pomardigital.net76.net");
        HttpURLConnection urlc = (HttpURLConnection)
        url.openConnection();
        urlc.setConnectTimeout(7000);
        urlc.connect();
        Log.i("VERIFICA", ""+urlc.getResponseCode());
        if (urlc.getResponseCode() == 200) {
            return new Boolean(true);
        }
    }
    catch (MalformedURLException e1) {
        e1.printStackTrace();
    }
    catch (IOException e) {
        e.printStackTrace();
    }
    }
    return false;
}

    public void VerificaLogin()
    {
        try {
            cursor = BD.query("users",
                new String [] {"IDUser", "Password"},
                "IDUser like '"+etuser.getText().toString()+"' and
                Password like '"+etpass.getText().toString()+"'", //faz busca pelas
                credenciais
                null, //selectionArgs,
                null, //groupBy,
                null, //having,
                null, //"order by nome" //orderBy
                null); // Limite de registros retornados

            int numeroRegistros = cursor.getCount();
            if (numeroRegistros != 0) // Se existirem registros na bd
            interna
            {
                id = etuser.getText().toString();
                login=true;
                if(isOnline()){
                    SincronizaServer();
                }
                main(); //Mostra a HomePage com as opcoes todas
            }
        }
    }
}

```



```

        }
        else
        {
            LoginServer(); // Função que executa a tarefa de
sincroniza o com o servidor

        }

    }
    catch (Exception erro) {
        msg("Erro Banco", "Erro buscar dados no banco:
"+erro.getMessage());

    }

}

public void chamascanner() {

    Intent intent= new
Intent("com.google.zxing.client.android.SCAN");
    intent.putExtra("SCAN_MODE","QR_CODE_MODE");
    startActivityForResult(intent,0);
    //O Intent é uma abstração descritiva da acção a ser
executada, faz basicamente uma chamada
    //a uma "actividade" presente noutro projecto, ou
seja, executa código presente noutro
    //projecto

}

public void arvore () {

    setContentView(R.layout.arvreginter);
    mViewMode=ViewMode.LERARVORES;
    TabHost th=(TabHost) findViewById(R.id.tabhost);
    th.setup();
    TabSpec tspec= th.newTabSpec("tag1");
    tspec.setContent(R.id.tab1);
    tspec.setIndicator("Criar Arvores");
    th.addTab(tspec);
    tspec= th.newTabSpec("tag2");
    tspec.setContent(R.id.tab2);
    tspec.setIndicator("Criar intervenções");
    th.addTab(tspec);
    if(tab1)
    {
        IDProdutor = (EditText) findViewById(R.id.IDProdutor);
        IDProdutor.setText(id);
        IDArvore = (EditText) findViewById(R.id.IDArvore);
        IDArvore.setText(idArvore);
        th.setCurrentTab(0);
    }
    else
    {
        txtIdProdInter=(TextView)
findViewById(R.intervencao.txtidProdutor);
        txtIdProdInter.setText("Id de Produtor:"+id);
        idArvoreIntervencao=(EditText)
findViewById(R.intervencao.edid);
        idArvoreIntervencao.setText(idArvoreInter);
        th.setCurrentTab(1);
    }
}

```

```

    }
    inicializaobjetos();
    listener();

}
public void login () {

    setContentView(R.layout.login);

    inicializaobjetos();
    listener();

}
public void msg(String titulo, String txt)
{
    AlertDialog.Builder msg = new AlertDialog.Builder(AppPomar.this);
    msg.setTitle(titulo);
    msg.setMessage(txt);
    msg.setNeutralButton("Ok", null);
    msg.show();
}
public void listener() {

    try{//pagina main
        if(login==true)
        {
            btlogin.setVisibility(Button.GONE);
            btArvoreMain.setVisibility(Button.VISIBLE);
            btConsulta.setVisibility(Button.VISIBLE);
            btEnvia.setVisibility(Button.VISIBLE);
        }
        else
        {
            btlogin.setOnClickListener(new View.OnClickListener()
            {

                @Override
                public void onClick(View v) {
                    login(); //chama p#na login
                }
            });
        }
        btsair.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(
                    context);

                // set title
                alertDialogBuilder.setTitle("Deseja
Sair?");

                // set dialog message
                alertDialogBuilder
                    .setMessage("Deseja sair da aplica#o?")
                    .setCancelable(false)

```

```

        .setPositiveButton("Sim",new
DialogInterface.OnClickListener() {
    public void
onClick(DialogInterface dialog,int id) {
        // if this button is clicked,
close
        // current activity
        AppPomar.this.finish();
    }
})
        .setNegativeButton("Nã",new
DialogInterface.OnClickListener() {
    public void
onClick(DialogInterface dialog,int id) {
        // if this button is clicked,
just close
        // the dialog box and do
nothing
        dialog.cancel();
    }
});

// create alert dialog
AlertDialog alertDialog =
alertDialogBuilder.create();

// show it
alertDialog.show();

    }
});

btArvoreMain.setOnClickListener(new
View.OnClickListener() {

    @Override
    public void onClick(View v) {
        mViewMode=ViewMode.LERARVORES;
        tab1=true;
        tab2=false;
        arvore();//chama arvore
    }
});

btEnvia.setOnClickListener(new View.OnClickListener()
{

    @Override
    public void onClick(View v) {
        AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(
            context);

        // set title

alertDialogBuilder.setTitle("Sincronizar");

        // set dialog message
        alertDialogBuilder

```

```

        .setMessage("Deseja sincronizar os
dados?")
        .setCancelable(false)
        .setPositiveButton("Sim", new
DialogInterface.OnClickListener() {
            public void
onClick(DialogInterface dialog, int id) {
                // if this button is clicked,
close
                // current activity
                try{
                    SincronizaServer();
                }
                catch(Exception erro){}
            }
        })
        .setNegativeButton("NÃo", new
DialogInterface.OnClickListener() {
            public void
onClick(DialogInterface dialog, int id) {
                // if this button is clicked,
just close
                // the dialog box and do
nothing
                dialog.cancel();
            }
        });

        // create alert dialog
AlertDialog alertDialog =
alertDialogBuilder.create();

        // show it
alertDialog.show();

    }
});

btWeb.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {

        if (isOnline()) {
            //Uri uriUrl =
Uri.parse("http://172.19.100.140/Exemplos/teste.php");
            Uri uriUrl =
Uri.parse("http://pomardigital.net76.net/default.php");
            Intent launchBrowser = new
Intent(Intent.ACTION_VIEW, uriUrl);
            startActivity(launchBrowser); //chama pÃgina web
        }
        else
        {
            msg("Falha", "Necessita de conecÃo
internet!");
        }
    }
});

```

```

    }
    catch(Exception erro)
    {}

    //Pagina Consulta Arvore
    try
    {
        btConsulta.setOnClickListner(new View.OnClickListener() {
            public void onClick(View v) {
                consultaTodas=true;
                tab1=false;
                tab2=false;
                chamaConsulta();
                inicializaobjetos();
                listener();
            }
        });
    }
    catch(Exception erro)
    {
    }

    try
    {
        btLerArvIdCons.setOnClickListner(new
View.OnClickListener() {
            public void onClick(View v) {
                cursor.close();
                chamascanner();
                listener();
            }
        });

        ImgBtIdPesquisa.setOnClickListner(new
View.OnClickListener() {
            public void onClick(View v) {
                chamaConsultaArvoreId();
                listener();
            }
        });
        btRegAnt.setOnClickListner(new View.OnClickListener() {
            public void onClick(View v) {
                mostrarRegistroAnterior();
            }
        });
    }
    catch(Exception erro)
    {
    }
    try
    {
        btProxReg.setOnClickListner(new View.OnClickListener() {
            public void onClick(View v) {
                mostrarProximoRegistro();
            }
        });
    }
    catch(Exception erro)
    {
    }
}

```

```

        }
    });
}
catch(Exception erro)
{
}

try
{
    btApagarArv.setOnClickListener(new View.OnClickListener()
    {
        public void onClick(View v) {

            AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(
                context);

            alertDialogBuilder.setTitle("Apagar?");

            alertDialogBuilder
                .setMessage("Deseja apagar este registro?
Esta opera玏 玀rreversl.")
                .setCancelable(false)
                .setPositiveButton("Sim",new
DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface
dialog,int id) {
                        String arvore;

                        arvore=txtid.getText().toString().substring(11);

                        DesactivaArvore(arvore.toString());

                        chamaConsulta();
                        listener();
                    }
                })
                .setNegativeButton("N玀",new
DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface
dialog,int id) {
                        dialog.cancel();
                    }
                });

            AlertDialog alertDialog =
alertDialogBuilder.create();

            alertDialog.show();

        }
    });
}
catch(Exception erro)
{
}

try
{
    btMenuP.setOnClickListener(new View.OnClickListener() {

```

```

        public void onClick(View v) {
            setContentView(R.layout.main);
            mViewMode=ViewMode.MAIN;
            inicializaobjetos();
            listener();
            cursor.close();
            //cursor_consulta.close();

        }
    });

}
catch(Exception erro)
{
}

//Consulta da arvore por interveniência
try{
    btLerConsInter.setOnClickListener(new
View.OnClickListener() {

        @Override
        public void onClick(View v) {
            //cursor.close();
            cursor_consulta.close();
            chamaScanner();
            listener();
        }
    });

    ImgBtPesquisa.setOnClickListener(new
View.OnClickListener() {
        public void onClick(View v) {
            chamaConsultaInter();
            listener();
        }
    });

    btRegAntInter.setOnClickListener(new
View.OnClickListener() {
        public void onClick(View v) {
            mostrarRegistoAnteriorInter();
        }
    });

    btProxRegInter.setOnClickListener(new
View.OnClickListener() {
        public void onClick(View v) {
            mostrarProximoRegistoInter();
        }
    });

    btApagarRegInter.setOnClickListener(new
View.OnClickListener() {
        public void onClick(View v) {
            AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(
                context);

```

```

        alertDialogBuilder.setTitle("Apagar?");

        alertDialogBuilder
            .setMessage("Deseja apagar este registo?
Esta operaçrreversl.")
            .setCancelable(false)
            .setPositiveButton("Sim",new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface
dialog,int id) {

                    String arvore,intervencao;

                    intervencao=txtIdConsInter.getText().toString().substring(19);
                    arvore=txtArvConsInter.getText().toString().substring(14);

                    DesactivaIntervencao(intervencao.toString(),arvore.toString());
                    chamaConsultaInter();
                    listener();

                }
            })
            .setNegativeButton("N",new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface
dialog,int id) {

                    dialog.cancel();

                }
            });

        // create alert dialog
        AlertDialog alertDialog =
alertDialogBuilder.create();

        // show it
        alertDialog.show();

    }

});

    btMenuPConsInter.setOnClickListener(new
View.OnClickListener() {

        @Override
        public void onClick(View v) {
            setContentView(R.layout.main);
            mViewMode=ViewMode.MAIN;
            inicializaobjectos();
            listener();

        }

    });

}
catch(Exception erro){

}

//*****Consulta da arvore por
Motivo*****
try
{
    btLerArvMotivo.setOnClickListener(new
View.OnClickListener() {

```



```

        @Override
        public void onClick(View v) {
            //cursor.close();
            cursor_consulta.close();
            chamascanner();
            listener();
        }
    });

    btRegAntMotivo.setOnClickListener(new
View.OnClickListener() {
        public void onClick(View v) {
            mostrarRegistoAnteriorMotivo();
        }
    });

    btProxRegMotivo.setOnClickListener(new
View.OnClickListener() {
        public void onClick(View v) {
            mostrarProximoRegistoMotivo();
        }
    });

    ImgBtPesquisaMotivo.setOnClickListener(new
View.OnClickListener() {
        public void onClick(View v) {
            if((tipo.toString().compareTo("Escolher:") == 0)) {
                msg("Aviso!", "Escolha um 'Motivo' para a
intervenç o");
            }
            else{
                chamaConsultaMotivoPorId(tipo);
                listener();
            }
        }
    });

    spMotivoCons.setOnItemSelectedListener(new
OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<?> parent,
View view, int position, long i)
        {
            if(parent.getItemAtPosition(position).toString().equalsIgnoreCase("Esc
olher:"))
            {
                chamaConsultaMotivoTodos();
            }
            else
            if(parent.getItemAtPosition(position).toString().equalsIgnoreCase("Col
heita"))
            {
                //Consulta por colheita

                chamaConsultaMotivo(parent.getItemAtPosition(position).toString());
                tipo="Colheita";
            }
        }
    });

```

```

        else
        if(parent.getItemAtPosition(position).toString().equalsIgnoreCase("Limpeza"))
        {
            //Consulta por limpeza

            chamaConsultaMotivo(parent.getItemAtPosition(position).toString());
            tipo="Limpeza";
        }
        else
        if(parent.getItemAtPosition(position).toString().equalsIgnoreCase("Tratamento")){
            //Consulta por tratamento

            chamaConsultaMotivo(parent.getItemAtPosition(position).toString());
            tipo="Tratamento";
        }
        else
        {
            chamaConsultaMotivo(parent.getItemAtPosition(position).toString());
            tipo="Outro";
        }
    }

    @Override
    public void onNothingSelected(AdapterView<?> arg0) {

    }

    });

    btMPmotivo.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            setContentView(R.layout.main);
            mViewMode=ViewMode.MAIN;
            inicializaobjetos();
            listener();
        }
    });

    btApagarMotivo.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            AlertDialog.Builder alertDialogBuilder = new
            AlertDialog.Builder(
                context);

            alertDialogBuilder.setTitle("Apagar?");

            alertDialogBuilder
                .setMessage("Deseja apagar este registro? Esta operac o   irrevers l.")
                .setCancelable(false)
                .setPositiveButton("Sim",new
            DialogInterface.OnClickListener() {

```

```

        public void onClick(DialogInterface
dialog,int id) {
            String arvore,intervencao;

            intervencao=txtIdConsInterMotivo.getText().toString().substring(19);
            arvore=txtArvConsMotivo.getText().toString().substring(14);

            DesactivaIntervencao(intervencao.toString(),arvore.toString());
                                chamaConsultaMotivo(tipo);
                                listener();
        }
    })
    .setNegativeButton("N\u00e3",new
DialogInterface.OnClickListener() {
        public void onClick(DialogInterface
dialog,int id) {
            dialog.cancel();
        }
    });

    // create alert dialog
    AlertDialog alertDialog =
alertDialogBuilder.create();

    // show it
    alertDialog.show();

    }
    });
}

catch(Exception erro){

}

//*****
try{//pagina login

    btsairlogin.setOnClickListener(new View.OnClickListener()
{

        @Override
        public void onClick(View v) {
            main(); //chama p\u00e1gina inicial
        }
    });

    btok.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            VerificaLogin();
        }
    });

}
}

```

```

        catch(Exception erro)
        {}

        try{//pagina arvore

            btarvore.setOnClickListener(new View.OnClickListener()

            {

                @Override
                public void onClick(View v) {
                    chamascanner();//chama leitor qrcode
                    tab2=false;
                    tab1=true;
                    consultaTodas=false;
                    arvore();//chama pagina arvore

                }

            });

            btguardar.setOnClickListener(new
View.OnClickListener() {

                @Override
                public void onClick(View v) {

                    InsertBD();//chama Insert

                }

            });

            btLimparArv.setOnClickListener(new
View.OnClickListener() {

                @Override
                public void onClick(View v) {
                    IDArvore.setText("");
                    inicializaobjetos();

                }

            });

            btvoltarmenu.setOnClickListener(new
View.OnClickListener() {

                @Override
                public void onClick(View v) {
                    main();

                }

            });

        }
        catch(Exception erro)
        {}

        try //pagina intervenções
        {

```

```

        btLerInter.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                chamascanner();//chama leitor qrcode
                tab1=false;
                tab2=true;
                consultaTodas=false;
                arvore();//chama pagina arvore

            }

        });

        btLimparInter.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                edQuantInter.setText("");
                idArvoreIntervencao.setText("");
                edObs.setText("");
                inicializaobjetos();

            }

        });

        btGravaInter.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                InsereIntervencao();

            }

        });

        btVoltarInter.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                main();

            }

        });

    }

    catch(Exception erro){}

}

@Override
public void onActivityResult(int requestCode, int resultCode, Intent intent)
{
    //gaqui que a magia acontece... Nesta "atividade" recebemos o
    resultado do scanner
    //e manipulamos como quisermos.
    if(requestCode==0)
    {
        if(resultCode==RESULT_OK)
        {
            String contents=intent.getStringExtra("SCAN_RESULT");
            if(tab1){
                idArvore=contents.toString();
                idArvoreInter="";
                idArvoreCons="";
                arvore();
            }
        }
    }
}

```

```

    }
    else if(consultaTodas){
        idArvoreCons=contents.toString();
        if(mainLayout.getVisibility()==View.VISIBLE)
        {
            edIdArv.setText(idArvoreCons);
            idArvoreInter="";
            idArvore="";
        }
        else
        if(intervencaoLayout.getVisibility()==View.VISIBLE){
            edIdInter.setText(idArvoreCons);
            idArvoreInter="";
            idArvore="";
        }
        else{
            edIdMotivo.setText(idArvoreCons);
            idArvoreInter="";
            idArvore="";
        }
    }
    else{
        idArvoreInter=contents.toString();
        idArvore="";
        idArvoreCons="";
        arvore();
    }
    idArvoreInter=contents.toString();
}
else if(resultCode==RESULT_CANCELED)
{
}
}
}

public void inicializaobjetos () {

    try{

        //main
        btlogin = (Button) findViewById(R.id.btlogin);
        btsair = (Button) findViewById(R.id.btsair);
        btWeb = (Button) findViewById(R.id.btWeb);
        btEnvia=(Button) findViewById(R.id.btEnvia);
        //login
        etuser = (EditText) findViewById(R.id.etuser);
        etpass = (EditText) findViewById(R.id.etpass);
        btok = (Button) findViewById(R.id.btok);
        btsairlogin = (Button) findViewById(R.id.btsairlogin);

        //Consulta;
        btArvoreMain=(Button) findViewById(R.id.btArvoreMain);
        btConsulta=(Button) findViewById(R.id.btConsulta);
        spinner_Consulta = (Spinner)
        findViewById(R.id.spinnerConsulta);
        spMotivoCons = (Spinner)
        findViewById(R.id.spinnerConsultaMotivo);
        try
        {

```

```

        btProxReg=(Button) findViewById(R.id.btProxReg);
        btMenuP=(Button) findViewById(R.id.btVoltar);
        btRegAnt=(Button) findViewById(R.id.btRegAnt);
        btLerArvIdCons=(Button) findViewById(R.id.lerArvId);

        ImgBtIdPesquisa=(ImageButton) findViewById(R.id.btPesquisaId);
        edIdArv=(EditText) findViewById(R.id.edidArvId);
        txtid=(TextView) findViewById(R.id.txtid);
        txtQR=(TextView) findViewById(R.id.txtQR);
        txtP=(TextView) findViewById(R.id.txtP);
        txtTipo=(TextView) findViewById(R.id.txtTipo);
        txtestado=(TextView) findViewById(R.id.txtestado);
        txtidProdutor=(TextView) findViewById(R.id.txtidProdutor);
        btApagarArv=(Button) findViewById(R.id.btapagarReg);
    }
    catch (Exception erro) {
        msg("Erro de inicializacao: ",erro.getMessage());
    }
    //*****Objectos CONSULTA
INTERVEN!O*****
    try{

        txtArvConsInter=(TextView) findViewById(R.consultainter.txtidArv);

        txtIdConsInter=(TextView) findViewById(R.consultainter.txtIdInter);

        txtDataInterCons=(TextView) findViewById(R.consultainter.txtData);

        txtMotivoInterCons=(TextView) findViewById(R.consultainter.txtMotivo);

        txtQuantInterCons=(TextView) findViewById(R.consultainter.txtQuantidade
);

        txtEstadoConsInter=(TextView) findViewById(R.consultainter.txtEstadoInt
erCons);

        txtIdProdConsInter=(TextView) findViewById(R.consultainter.txtIdProdCon
sInter);

        txtObsInter=(TextView) findViewById(R.consultainter.txtObs);

        btProxRegInter=(Button) findViewById(R.consultainter.btProxReg);

        btRegAntInter=(Button) findViewById(R.consultainter.btRegAnt);

        btApagarRegInter=(Button) findViewById(R.consultainter.btApagarReg);

        btLerConsInter=(Button) findViewById(R.consultainter.lerArvore);

        btMenuPConsInter=(Button) findViewById(R.consultainter.btVoltar);

        edIdInter=(EditText) findViewById(R.consultainter.edidInter);

        ImgBtPesquisa=(ImageButton) findViewById(R.consultainter.btPesquisa);

        mainLayout=(LinearLayout) this.findViewById(R.id.linearLayoutmain);

        intervencaoLayout=(LinearLayout) this.findViewById(R.id.linearLayoutInt
er);

```

```

motivoLayout=(LinearLayout) this.findViewById(R.id.linearLayoutInterMotivo);

        ArrayAdapter<CharSequence> adapter_consulta =
ArrayAdapter.createFromResource(
            this, R.array.ListaFiltros,
            android.R.layout.simple_spinner_item);

adapter_consulta.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spinner_consulta.setAdapter(adapter_consulta);
        spinner_consulta.setPrompt("Filtrar Consulta
por:");

        spinner_consulta.setOnItemSelectedListener(new
OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?>
parent, View view, int position, long id)
            {

if(parent.getItemAtPosition(position).toString().equalsIgnoreCase("Intervenç es"))
            {
                //cursor.close();
                chamaConsultaInterTodas();

motivoLayout.setVisibility(LinearLayout.GONE);

            }
            else
if(parent.getItemAtPosition(position).toString().equalsIgnoreCase("Motivo"))
            {

mainLayout.setVisibility(LinearLayout.GONE);

intervencaoLayout.setVisibility(LinearLayout.GONE);

motivoLayout.setVisibility(LinearLayout.VISIBLE);

            }
            else
            {

mainLayout.setVisibility(LinearLayout.VISIBLE);

intervencaoLayout.setVisibility(LinearLayout.GONE);

motivoLayout.setVisibility(LinearLayout.GONE);

            }

            }

            @Override
            public void
onNothingSelected(AdapterView<?> arg0) {

            }
        }
    }
}

```



```

        });

    }
    catch (Exception erro) {

    }
    //*****FIM OBJETOS CONSULTA
INTERVEN!O*****

    //*****OBJETOS CONSULTA
MOTIVO*****

    try
    {

txtArvConsMotivo=(TextView) findViewById(R.consultaintermotivo.txtidArv
);

txtIdConsInterMotivo=(TextView) findViewById(R.consultaintermotivo.txtI
dInter);

txtDataInterMotivo=(TextView) findViewById(R.consultaintermotivo.txtDat
a);

txtQuantConsMotivo=(TextView) findViewById(R.consultaintermotivo.txtQua
ntidade);

txtEstadoConsInterMotivo=(TextView) findViewById(R.consultaintermotivo.
txtEstadoInterCons);

txtIdProdConsMotivo=(TextView) findViewById(R.consultaintermotivo.txtId
ProdConsInter);

txtMotivoInterConsMotivo=(TextView) findViewById(R.consultaintermotivo.
txtMotivo);

txtObsMotivo=(TextView) findViewById(R.consultaintermotivo.txtObs);

btRegAntMotivo=(Button) findViewById(R.consultaintermotivo.btRegAnt);

btProxRegMotivo=(Button) findViewById(R.consultaintermotivo.btProxReg);

btApagarMotivo=(Button) findViewById(R.consultaintermotivo.btApagarReg)
;

btLerArvMotivo=(Button) findViewById(R.consultaintermotivo.lerArvore);

ImgBtPesquisaMotivo=(ImageButton) findViewById(R.consultaintermotivo.bt
Pesquisa);

btMPmotivo=(Button) findViewById(R.consultaintermotivo.btVoltar);

edIdMotivo=(EditText) findViewById(R.consultaintermotivo.edidInter);

        ArrayAdapter<CharSequence> adapter_consulta_motivo =
ArrayAdapter.createFromResource(
            this, R.array.ListaMotivos,
            android.R.layout.simple_spinner_item);

        adapter_consulta_motivo.setDropDownViewResource(android.R.layout.simpl
e_spinner_dropdown_item);
    }
}

```

```

spMotivoCons.setAdapter(adapter_consulta_motivo);
spMotivoCons.setPrompt("Filtrar Motivo por:");

    }
    catch (Exception erro)
    {

    }
    //*****FIM OBJECTOS CONSULTA
MOTIVO*****

//*****Intervencao*****
*****

        btGravaInter
        =(Button) findViewById(R.intervencao.btgravar);

        btVoltarInter=(Button) findViewById(R.intervencao.btvoltarmenu);
        btLerInter=(Button) findViewById(R.intervencao.btler);

        btLimparInter=(Button) findViewById(R.intervencao.btproximo);

        btProximoInter=(Button) findViewById(R.intervencao.btproximo);

        txtIdProdInter=(TextView) findViewById(R.intervencao.txtidProdutor);

        txtQuantInter=(TextView) findViewById(R.intervencao.txtQuantidade);

        edQuantInter=(EditText) findViewById(R.intervencao.edQuant);

        dataIntervencao=(DatePicker) findViewById(R.intervencao.dataIntervencao
        );

        idArvoreIntervencao=(EditText) findViewById(R.intervencao.edid);

        txtObservacao=(TextView) findViewById(R.intervencao.txtObs);
        edObs=(EditText) findViewById(R.intervencao.edObs);
        spinner_motivo = (Spinner)
        findViewById(R.intervencao.motivo);
        ArrayAdapter<CharSequence> adapter_motivo =
        ArrayAdapter.createFromResource(
            this, R.array.ListaMotivos,
            android.R.layout.simple_spinner_item);

        adapter_motivo.setDropDownViewResource(android.R.layout.simple_spinner
        _dropdown_item);
        spinner_motivo.setAdapter(adapter_motivo);
        spinner_motivo.setPrompt("Escolha o motivo");
        spinner_motivo.setOnItemClickListener(new
        OnItemSelectedListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View
            view, int position, long i)
            {

                if(parent.getItemAtPosition(position).toString().equalsIgnoreCase("Col
                heita"))
                {

```

```

        edQuantInter.setVisibility(EditText.VISIBLE);
        txtQuantInter.setVisibility(TextView.VISIBLE);
        edObs.setVisibility(EditText.VISIBLE);
        txtObservacao.setVisibility(TextView.VISIBLE);
        tipo="Colheita";
    }
    else
    {
        edQuantInter.setVisibility(EditText.GONE);
        txtQuantInter.setVisibility(TextView.GONE);
        edObs.setVisibility(EditText.VISIBLE);
        txtObservacao.setVisibility(TextView.VISIBLE);
    }

    if(parent.getItemAtPosition(position).toString().equalsIgnoreCase("Limpeza"))
    {
        tipo="Limpeza";
    }

    if(parent.getItemAtPosition(position).toString().equalsIgnoreCase("Tratamento"))
    {
        tipo="Tratamento";
    }

    if(parent.getItemAtPosition(position).toString().equalsIgnoreCase("Outro"))
    {
        tipo="Outro";
    }

    if(parent.getItemAtPosition(position).toString().equalsIgnoreCase("Escolher:"))
    {
        tipo="Escolher:";
    }

    motivo=parent.getItemAtPosition(position).toString();
}

@Override
public void onNothingSelected(AdapterView<?> arg0)
{

}

});

//arvore
btvoltarmenu = (Button) findViewById(R.id.btvoltarmenu);
btguardar = (Button) findViewById(R.id.btguardar);
btarvore = (Button) findViewById(R.id.btarvore);
IDArvore = (EditText) findViewById(R.id.IDArvore);
tipoarvore = (Spinner) findViewById(R.id.spinnertipo);
dataQ=(DatePicker) findViewById(R.id.datePicker2);
dataP=(DatePicker) findViewById(R.id.datePicker1);
btLimparArv=(Button) findViewById(R.id.btleitura);
Spinner spinner = (Spinner)
findViewById(R.id.spinnertipo);

```

```

        ArrayAdapter<CharSequence> adapter =
ArrayAdapter.createFromResource(
            this, R.array.ListaArvores,
            android.R.layout.simple_spinner_item);

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdo
wn_item);

        spinner.setAdapter(adapter);
        spinner.setPrompt("Tipo de Árvore");

    }

    catch(Exception erro)
    {}
}

public void OpenCreateBD(){
    String nomeBD = "pomar";
    try
    {

        //cria ou abre BD
        BD = openOrCreateDatabase(nomeBD, MODE_WORLD_READABLE, null);
        String sql = "CREATE TABLE IF NOT EXISTS arvore "
            + "(idProc INTEGER PRIMARY KEY, IDArvore TEXT,
Data_Plantacao TEXT, "
            + "Data_Implementacao TEXT, Tipo_Arvore TEXT,
estado TEXT, idProdutor TEXT, inativo TEXT);";
        BD.execSQL(sql);
        //msg("BD", "BD criada com sucesso!! ");
    }
    catch (Exception erro)
    {

        msg("BD", "Erro ao criar ou abrir a BD");
    }

    try
    {

        //cria ou abre BD
        BD = openOrCreateDatabase(nomeBD, MODE_WORLD_READABLE, null);
        String sql = "CREATE TABLE IF NOT EXISTS users "
            + "(IDUser TEXT PRIMARY KEY, Password TEXT);";
        BD.execSQL(sql);
        //msg("BD", "BD criada com sucesso!! ");
    }
    catch (Exception erro)
    {

        msg("BD", "Erro ao criar ou abrir a BD");
    }

    try
    {

        //cria ou abre BD
        BD = openOrCreateDatabase(nomeBD, MODE_WORLD_READABLE, null);
        String sql = "CREATE TABLE IF NOT EXISTS intervencoes "
            + "(idIntervencao TEXT PRIMARY KEY,
data_intervencao TEXT, idArvore TEXT, Motivo TEXT, idProdutor TEXT,
quantidade TEXT NULL, observacao TEXT NULL, estado TEXT, inativo
TEXT);";
    }
}

```

```

        BD.execSQL(sql);
        //msg("BD", "BD criada com sucesso!! ");
    }
    catch(Exception erro)
    {
        msg("BD", "Erro ao criar ou abrir a BD");
    }

    /*try
    {
        //cria ou abre BD
        BD = openOrCreateDatabase(nomeBD, MODE_WORLD_READABLE, null);
        String sql = "CREATE TABLE IF NOT EXISTS Motivo "
            + "(Motivo Integer PRIMARY KEY, descricao TEXT,
quantidade Integer, idIntervencao Integer);";
        BD.execSQL(sql);
        //msg("BD", "BD criada com sucesso!! ");
    }
    catch(Exception erro)
    {
        msg("BD", "Erro ao criar ou abrir a BD");
    }
    */

}
public boolean isInteger( String input )
{
    try
    {
        Integer.parseInt( input );
        return true;
    }
    catch( Exception erro)
    {
        return false;
    }
}

public void InsertBD(){
    String estado="0";
    String inativo="0";
    Calendar c = Calendar.getInstance();
    SimpleDateFormat df = new SimpleDateFormat("yyyyMMdd");
    String data = df.format(c.getTime());
    try{
        df.getCalendar();

qrdata=dataQ.getDayOfMonth()+"/"+dataQ.getMonth()+"/"+dataQ.getYear();

plantacao=dataP.getDayOfMonth()+"/"+dataP.getMonth()+"/"+dataP.getYear
();
        int ano=Integer.parseInt(data.substring(0,4));
        int mes=Integer.parseInt(data.substring(4,6));
        int dia=Integer.parseInt(data.substring(6,8));

if((((dataQ.getDayOfMonth()<=dataP.getDayOfMonth()) && (mesQ<=mesP) && (da
taQ.getYear()<=dataP.getYear())) || ((mesQ<mesP) &&
(dataQ.getYear()<=dataP.getYear())) ||
(dataQ.getYear()<dataP.getYear()) || (dataQ.getDayOfMonth()>dia) &&
(mesQ>=mes) && (dataQ.getYear()>=ano)) || (mesQ>mes) &&
(dataQ.getYear()>=ano) || ((dataQ.getYear()>ano)) || ((dataP.getDayOfMonth
())>dia) && (mesP>=mes) && (dataP.getYear()>=ano)) || (mesP>mes) &&
(dataP.getYear()>=ano) || ((dataP.getYear()>ano)))
        {

```

```

        if((IDArvore.getText().toString().compareTo("")!=0))
        {
            cursor = BD.query("arvore",
                                new String []
{"IDArvore","Data_Plantacao","Data_Implementacao","Tipo_Arvore","idPro
dutor"},
                                "IDArvore like
'" +IDArvore.getText().toString()+"' and inativo like '0'",//estado
like '%0%',//selection,//estado like '%0%',//selection,
                                null,//selectionArgs,
                                null,//groupBy,
                                null,//having,
                                null,//"order by nome"//orderBy)
                                null); // Limite de registros retornados
            int numeroRegistros = cursor.getCount();
            if (numeroRegistros != 0)
            {
                msg("Aviso!", "Essa árvore já existe e está ativa!");
            }
            else
            {

if(tipoarvore.getSelectedItem().toString().compareTo("Escolher:")!=0)
{
                String sql = "INSERT INTO arvore (IDArvore,
Data_Plantacao, Data_Implementacao, Tipo_Arvore, estado,
idProdutor,inativo) values ( '"
                    +IDArvore.getText().toString()+"', '"
                    +plantacao.toString()+"', '"
                    +qrdata.toString()+"', '"

+tipoarvore.getSelectedItem().toString()+/*.getSelectedItemPosition()*/
+"', '"

                    +estado.toString()+"', '"
                    +id.toString()+"', '"
                    +inativo.toString()+"')";

                Log.i("BD", "entrou no evento"+sql); //->LogCat

                BD.execSQL(sql);

                msg("Conclu!", "Dados gravados com sucesso!");
            }
            else
            {
                msg("Aviso!", "Escolha o tipo de árvore!");
            }
        }
    }
    else
    {
        msg("Aviso!", "Falta ler o id da árvore!");
    }
}
else{msg("Aviso!", "Existem dados inválidos! *verifique as
datas*");}

}
catch (Exception erro)
{

```

```

        msg("Erro BD", "Não foi possível guardar dados!" + erro);

        erro.printStackTrace();
    }
}

public void InsereIntervencao()
{
    intervencao = dataIntervencao.getDayOfMonth() + "/" + dataIntervencao.getMonth() + "/" + dataIntervencao.getYear();
    String estado = "0";
    String inativo = "0";
    Calendar c = Calendar.getInstance();
    SimpleDateFormat df = new SimpleDateFormat("yyyyMMddHHmmss");
    String data = df.format(c.getTime());
    String idInter;

    idInter = id.toString() + "" + idArvoreInter.toString() + "" + data.toString();

    try {
        df.getCalendar();
        int ano = Integer.parseInt(data.substring(0, 4));
        int mes = Integer.parseInt(data.substring(4, 6));
        int dia = Integer.parseInt(data.substring(6, 8));

        if((idArvoreIntervencao.getText().toString().compareTo("") != 0) &&
            if((dataIntervencao.getDayOfMonth() >= dia &&
                (dataIntervencao.getMonth() <= mes &&
                dataIntervencao.getYear() <= ano)) || (dataIntervencao.getDayOfMonth() <= dia &&
                (dataIntervencao.getMonth() <= mes &&
                dataIntervencao.getYear() <= ano)) || (dataIntervencao.getDayOfMonth() >= dia &&
                (dataIntervencao.getMonth() >= mes &&
                dataIntervencao.getYear() <= ano))) {

            cursor = BD.query("arvore",
                new String []
            {"IDArvore", "Data_Plantacao", "Data_Implementacao", "Tipo_Arvore", "idProdutor"},

                "IDArvore like '" + idArvoreIntervencao.getText().toString() + "' and idProdutor like '" + id.toString() + "' and inativo like '0'", // "estado like '%0%' ", // "selection, // "estado like '%0%' ", // "selection,
                null, // "selectionArgs,
                null, // "groupBy,
                null, // "having,
                null, // "order by nome" // "orderBy)
                null); // Limite de registros retornados
            int numeroRegistros = cursor.getCount();
            if (numeroRegistros != 0)
            {
                if((tipo.toString().compareTo("Escolher:") == 0)) {
                    msg("Aviso!", "Escolha um 'Motivo' para a intervenção");
                }
                else {
                    if((isInteger(edQuantInter.getText().toString())))
                    {
                        Log.i("TIPO", tipo.toString());
                    }
                }
            }
        }
    }
}

```

```

        String sql="INSERT INTO
intervencoes(idIntervencao,data_intervencao, idArvore, Motivo,
idProdutor, quantidade, observacao, estado, inativo) values('"
        +idInter.toString()+"', '"
        +intervencao.toString()+"', '"
        +idArvoreInter.toString()+"', '"
        +motivo.toString()+"', '"
        +id.toString()+"', '"
        +edQuantInter.getText().toString()+"', '"
        +edObs.getText().toString()+"', '"
        +estado.toString()+"', '"
        +inativo.toString()+"'");

        BD.execSQL(sql);

        msg("Conclu!", "Dados gravados com sucesso!");
    }
    else if((edQuantInter.getVisibility()==View.GONE))
    {
        String sql="INSERT INTO
intervencoes(idIntervencao,data_intervencao, idArvore, Motivo,
idProdutor, quantidade, observacao, estado, inativo) values('"
        +idInter.toString()+"', '"
        +intervencao.toString()+"', '"
        +idArvoreInter.toString()+"', '"
        +motivo.toString()+"', '"
        +id.toString()+"', '"
        +id.toString()+"', '"
        +edObs.getText().toString()+"', '"
        +estado.toString()+"', '"
        +inativo.toString()+"'");

        BD.execSQL(sql);

        msg("Conclu!", "Dados gravados com sucesso!");
    }
    else
    if(edQuantInter.getText().toString().compareTo("")==0)
    {
        msg("Aviso!", "Arvore nªo registrada. Verifique o campo
'Quantidade'");
    }

    else
        msg("Aviso!", "Arvore nªo registrada. Verifique os
campos inseridos");
    }
    else
    {
        msg("Aviso!", "Arvore nªo presente no dispositivo.");
    }
    }
    else{msg("Aviso!", "Existem dados inválidos! *verifique a
data*");}
    }
    else
    {msg("Aviso!", "Falta o id da Arvore.");}
    }
    catch(Exception erro){}
}

```



```

public void InsereUser()
{
    try{

        String sql = "INSERT INTO users( IDUser,Password) values (
            "+etuser.getText().toString()+"', '"+
            +etpass.getText().toString()+"')";

        Log.i("BD", "entrou no evento"+sql); //->LogCat

        BD.execSQL(sql);

        //msg("Conclu!", "Dados gravados com sucesso!");

    }
    catch (Exception erro)
    {
        //msg("Erro BD", "Nã foi possl guardar dados!" + erro);
        erro.printStackTrace();
    }
}

public boolean ConsultaArvores()
{
    try {
        cursor = BD.query("arvore",
            new String []
{"IDArvore", "Data_Plantacao", "Data_Implementacao", "Tipo_Arvore", "estad
o", "idProdutor"},
            "idProdutor like '"+id.toString()+"' and
inativo like '0'", //"estado like '%0%'", //selection, null, //"estado
like '%0%'", //selection,
            null, //selectionArgs,
            null, //groupBy,
            null, //having,
            "idArvore", //"order by nome" //orderBy
            null); // Limite de registros retornados
        id_arvore=cursor.getColumnIndex("IDArvore");
        data_plantacao=cursor.getColumnIndex("Data_Plantacao");

        data_scanner=cursor.getColumnIndex("Data_Implementacao");
        tipo_arvore=cursor.getColumnIndex("Tipo_Arvore");
        estado_CA=cursor.getColumnIndex("estado");
        id_Produtor=cursor.getColumnIndex("idProdutor");
        int numeroRegistros = cursor.getCount();
        if (numeroRegistros != 0)
        {
            // no java puro resultsetwt.first();
            cursor.moveToFirst(); //posiciona no primeiro
registro

            return true;
        }
        else
            return false;

    }
    catch (Exception erro) {
        msg("Erro Banco", "Erro buscar dados no banco:
"+erro.getMessage());
        return false;
    }
}

```

```

    }
}

public boolean ConsultaArvoresId()
{
    try {
        cursor = BD.query("arvore",
            new String []
{"IDArvore","Data_Plantacao","Data_Implementacao","Tipo_Arvore","estado",
"IDProdutor"},
            "idProdutor like '"+id.toString()+"' and IDArvore
like '"+idArvoreCons.toString()+"' and inativo like '0'",// "estado
like '%0%'",//selection,null,// "estado like '%0%'",//selection,
            null, //selectionArgs,
            null, //groupBy,
            null, //having,
            null, // "order by nome" //orderBy
            null); // Limite de registros retornados
        id_arvore=cursor.getColumnIndex("IDArvore");
        data_plantacao=cursor.getColumnIndex("Data_Plantacao");
        data_scanner=cursor.getColumnIndex("Data_Implementacao");
        tipo_arvore=cursor.getColumnIndex("Tipo_Arvore");
        estado_CA=cursor.getColumnIndex("estado");
        id_Produtor=cursor.getColumnIndex("idProdutor");
        int numeroRegistros = cursor.getCount();
        if (numeroRegistros != 0)
        {
            // no java puro resultsewt.first();
            cursor.moveToFirst(); //posiciona no primeiro registro
            return true;
        }
        else
            return false;

    }
    catch(Exception erro) {
        msg("Erro Banco", "Erro buscar dados no banco:
"+erro.getMessage());
        return false;
    }
}

public void chamaConsultaArvoreId()
{
    if (ConsultaArvoresId())
    {
        setContentView(R.layout.consulta_arvore);
        mViewMode = ViewMode.CONSULTAS;
        inicializaobjectos();
        listener();
        mostrarDados();
    }
    else
    {
        msg("Aviso!", "Nãopossui registros gravados");
        setContentView(R.layout.main);
        inicializaobjectos();
        listener();
        return;
    }
}
}

```

```

    public void mostrarDados() {
        //dados de consulta da floresta
        txtid.setText("id Arvore: "+ cursor.getString(id_arvore));
        txtP.setText("Data de Plantação:
"+cursor.getString(data_plantacao));
        txtQR.setText("Data de implementação:
"+cursor.getString(data_scanner));
        txtTipo.setText("Tipo de floresta: "+cursor.getString(tipo_arvore));
        txtestado.setText("");
        txtidProdutor.setText("");
    }

    public void mostrarDadosInter(){

        txtArvConsInter.setText("Id da floresta:
"+cursor_consulta.getString(id_arvoreCons));
        txtIdConsInter.setText("Id da intervenção:
"+cursor_consulta.getString(idInterCons));
        txtDataInterCons.setText("Data da intervenção:
"+cursor_consulta.getString(data_scanner));
        txtMotivoInterCons.setText("Motivo da intervenção:
"+cursor_consulta.getString(motivoInter));
        txtQuantInterCons.setText("Quantidade recolhida (kg):
"+cursor_consulta.getString(quantidade));

        txtEstadoConsInter.setText("Estado:"+cursor_consulta.getString(estadosConsInter));
        txtIdProdConsInter.setText("id
Produtor:"+cursor_consulta.getString(idProdutorConsInter));
        txtObsInter.setText("Observação:
"+cursor_consulta.getString(obsCons));
        btRegAntInter.setVisibility(Button.VISIBLE);
        btProxRegInter.setVisibility(Button.VISIBLE);
        if(cursor_consulta.getString(quantidade).compareTo("")==0)
        {
            txtQuantInterCons.setVisibility(View.GONE);
        }
        else
        {
            txtQuantInterCons.setVisibility(View.VISIBLE);
        }
    }

    public void chamaConsulta()
    {
        if (ConsultaArvores())
        {
            setContentView(R.layout.consulta_arvore);
            mViewMode = ViewMode.CONULTAS;
            inicializaobjectos();
            listener();
            mostrarDados();
        }
        else
        {
            msg("Aviso!", "Não possui registos gravados");
            setContentView(R.layout.main);
            inicializaobjectos();
        }
    }

```

```

        listener();
        return;
    }
}

public void chamaConsultaInter()
{
    if (ConsultaArvoreIntervencao())
    {
        listener();
        mostrarDadosInter();

    }
    else
    {
        msg("Aviso!", "Não possui registros gravados");
        chamaConsultaInterTodas();
        listener();
        return;
    }
}

public void mostrarRegistroAnterior()
{
    //btRegAnt.setOnClickListener(new View.OnClickListener() {

        //public void onClick(View v) {
        try {
            cursor.moveToPrevious();
            mostrarDados();

        }
        catch (Exception erro) {
            msg("Navegacao", "Você está no primeiro registro");
        }
        //}
        //});

}

public void mostrarProximoRegistro() {

    btProxReg.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            try {
                cursor.moveToNext();
                mostrarDados();

            }
            catch (Exception erro) {
                msg("Navegacao", "Você está no último registro");
            }
        }
    });
}

public void mostrarProximoRegistroInter()
{
    //btProxRegInter.setOnClickListener(new View.OnClickListener() {
    //    public void onClick(View v) {
    try {
        cursor_consulta.moveToNext();
        mostrarDadosInter();

    }
    catch (Exception erro) {

```

```

        msg("Navegacao", "Você está no último registro");
    }
    // }
    // });
}
public void mostrarRegistroAnteriorInter()
{
    // btRegAntInter.setOnClickListener(new View.OnClickListener() {

        //public void onClick(View v) {
        try {
            cursor_consulta.moveToPrevious();
            mostrarDadosInter();

        }
        catch(Exception erro) {
            msg("Navegacao", "Você está no primeiro registro");
        }
        // }
        // });

}

public void mostrarProximoRegistroMotivo()
{
    // btProxRegMotivo.setOnClickListener(new View.OnClickListener() {

        // public void onClick(View v) {
        try {
            cursor_consulta.moveToNext();
            mostrarDadosInterMotivo();

        }
        catch(Exception erro) {
            msg("Navegacao", "Você está no último registro");
        }
        // }
        // });

}

public void mostrarRegistroAnteriorMotivo()
{
    // btRegAntMotivo.setOnClickListener(new View.OnClickListener() {

        // public void onClick(View v) {
        try {
            cursor_consulta.moveToPrevious();
            mostrarDadosInterMotivo();

        }
        catch(Exception erro) {
            msg("Navegacao", "Você está no primeiro registro");
        }
        // }
        // });

}

public void EnviaHTTP(JSONArray jsonarray,String url)
{
    try {
        HttpParams httpParams = new BasicHttpParams();
        HttpConnectionParams.setConnectionTimeout(httpParams,
            TIMEOUT_MILLISEC);
    }
}

```

```

        HttpConnectionParams.setSoTimeout(httpParams,
TIMEOUT_MILLISEC);
        // Create a new HttpClient and Post Header
        HttpClient httpclient = new DefaultHttpClient();
        //HttpPost httpppost = new
HttpPost("http://172.19.100.140/RegistaArvore.php");
        HttpPost httpppost = new HttpPost(url);
        // Post the data:
        //StringEntity se = new StringEntity(jsonarray.toString());
        httpppost.setEntity(new
ByteArrayEntity(jsonarray.toString().getBytes("UTF8")));
        httpppost.setHeader("json", "application/json");
        // Execute HTTP Post Request
        HttpResponse response = httpclient.execute(httpppost);
        // for JSON:
        HttpEntity entity = response.getEntity();
        if (entity != null) {
            InputStream instream = entity.getContent();
            String result = RestClient.convertStreamToString(instream);
            Log.i("Read from server", result);
        }
    }
    catch (ClientProtocolException erro){}
    catch (IOException e) {
    }
}
public void EnviaDados(String url)
{
    try {

        cursor = BD.query("arvore",
            new String []
{"IDArvore", "Data_Plantacao", "Data_Implementacao", "Tipo_Arvore", "idPro
dutor", "inativo"},
            "estado like '0' and
idProdutor='"+id.toString()+"'", //selection,
            null, //selectionArgs,
            null, //groupBy,
            null, //having,
            null, //"order by nome" //orderBy
            null); // Limite de registros retornados
        startManagingCursor(cursor);

        if(cursor!=null)
        {
            cursor.moveToFirst();
            JSONArray jsonarray=new JSONArray();
            while (cursor.isAfterLast() == false) {

                try {
                    JSONObject json= new JSONObject();

                    json.put("IDArvore", cursor.getString(cursor.getColumnIndex("IDArvore")
));

                    json.put("Data_Plantacao", cursor.getString(cursor.getColumnIndex("Data
_Plantacao")));

```

```

        json.put("Data_Implementacao", cursor.getString(cursor.getColumnIndex("Data_Implementacao")));

        json.put("Tipo_Arvore", cursor.getString(cursor.getColumnIndex("Tipo_Arvore")));

        json.put("idProdutor", cursor.getString(cursor.getColumnIndex("idProdutor")));

        json.put("inativo", cursor.getString(cursor.getColumnIndex("inativo")));
    }

    jsonArray.put(json);

    cursor.moveToNext();
}
catch (Exception e) {

}
}
EnviaHTTP(jsonArray, url);
}
cursor.close();
updateDadosArvore();

}
catch (Exception erro) {
    //msg("Erro Banco", "Erro buscar dados no banco:
    "+erro.getMessage());
}

}

public void EnviaDadosIntervencao(String url)
{

    try {

        cursor = BD.query("intervencoes",
            new String []
{"idIntervencao", "data_intervencao", "idArvore", "Motivo", "idProdutor", "
quantidade", "observacao", "inativo"},
            "estado like '0' and
idProdutor='"+id.toString()+"'", //selection,
            null, //selectionArgs,
            null, //groupBy,
            null, //having,
            null, //"order by nome" //orderBy)
            null); // Limite de registros retornados
        startManagingCursor(cursor);
        if (cursor != null)
        {
            cursor.moveToFirst();
            JSONArray jsonArray = new JSONArray();
            while (cursor.isAfterLast() == false) {

                try {

```

```

        JSONObject json= new JSONObject();

        json.put("num_intervencao",cursor.getString(cursor.getColumnIndex("idI
ntervencao")));

        json.put("Data_Intervencao",cursor.getString(cursor.getColumnIndex("da
ta_intervencao")));

        json.put("Arvores_Produtor_idProdutor",cursor.getString(cursor.getColu
mnIndex("idProdutor")));

        json.put("Arvores_idArvore",cursor.getString(cursor.getColumnIndex("id
Arvore")));

        json.put("Motivo",cursor.getString(cursor.getColumnIndex("Motivo")));

        json.put("Quantidade",cursor.getString(cursor.getColumnIndex("quantida
de")));

        json.put("observacao",cursor.getString(cursor.getColumnIndex("observac
ao")));

        json.put("inativo",cursor.getString(cursor.getColumnIndex("inativo")))
;

        jsonarray.put(json);

        cursor.moveToNext();
    }
    catch (Exception e) {

    }

}

EnviaHTTP(jsonarray,url);
}

cursor.close();
updateDadosIntervencao();
}
catch(Exception erro) {
    //msg("Erro Banco", "Erro buscar dados no banco:
"+erro.getMessage());
}

}

public void updateDadosArvore()
{
    String estado="1";
    try
    {
        String sql="UPDATE arvore SET estado='"+estado.toString()+"'
Where idProdutor='"+id.toString()+"'";

```



```

        BD.execSQL(sql);
        String sql2="DELETE FROM arvore Where inativo='1'";
        BD.execSQL(sql2);
    }
    catch(Exception erro)
    {

    }
}
public void updateDadosIntervencao()
{
    String estado="1";
    try
    {
        String sql="UPDATE intervencoes SET
estado='"+estado.toString()+"' Where idProdutor='"+id.toString()+"'";
        BD.execSQL(sql);
        String sql2="DELETE FROM intervencoes Where inativo='1'";
        BD.execSQL(sql2);
    }
    catch(Exception erro)
    {

    }
}

public void recebeDados(String urlPost)
{
    String result="";

    result=SqlHttpHelper(urlPost);
    try{
        JSONArray jArray = new JSONArray(result);
        for(int i=0;i<jArray.length();i++){
            JSONObject json_data = jArray.getJSONObject(i);
            String query= "insert or replace into arvore
(IDArvore,Data_Plantacao,Data_Implementacao,Tipo_Arvore,estado,idProdu
tor,inativo)
values('"+json_data.getString("numArvore")+",'"+json_data.getString("
Data_Plantacao")+",'"+json_data.getString("Data_Implementacao")+",'"+
json_data.getString("Tipo_Arvore")+",'1','"+json_data.getString("Pro
dutor_idProdutor")+",'"+json_data.getString("inativo")+')";
            BD.execSQL(query);
            Log.i("ANTES DE APAGAR","");
        }
    }
    catch(Exception erro){

    }

}

public void recebeDadosAtualizados(String urlPost)
{
    String result="";

    result=SqlHttpHelper(urlPost);
    try{
        JSONArray jArray = new JSONArray(result);

```

```

        for(int i=0;i<jArray.length();i++){
            JSONObject json_data = jArray.getJSONObject(i);
            String query= "Update arvore set
IDArvore='"+json_data.getString("numArvore")+"',Data_Plantacao='"+json
_data.getString("Data_Plantacao")+"',Data_Implementacao='"+json_data.g
etString("Data_Implementacao")+"',Tipo_Arvore='"+json_data.getString("
Tipo_Arvore")+"',estado='1',idProdutor='"+json_data.getString("Produto
r_idProdutor")+"',inativo='"+json_data.getString("inativo")+" where
IDArvore='"+json_data.getString("numArvore")+" and estado='1'";
            BD.execSQL(query);
            Log.i("ANTES DE APAGAR","");
        }
    }
    catch(Exception erro){

    }

}

public void recebeDadosApagados(String urlPost)
{
    String result="";

    result=SqlHttpHelper(urlPost);
    try{
        JSONArray jArray = new JSONArray(result);
        for(int i=0;i<jArray.length();i++){
            JSONObject json_data = jArray.getJSONObject(i);
            String query= "DELETE FROM arvore where
IDArvore='"+json_data.getString("numArvore")+" and inativo='0'";
            BD.execSQL(query);
            Log.i("ANTES DE APAGAR","");
        }
    }
    catch(Exception erro){

    }
}

public String SqlHttpHelper(String urlPost)
{
    String result = "";
    HttpResponse response=null;
    InputStream is=null;
    String lastsync = null;
    //http get

    if (isOnline()) {

        try{
            //lastsync = URLEncoder.encode(lastsync,"utf-8");
            String address = urlPost + "?user="+id.toString();
            HttpClient httpclient = new DefaultHttpClient();
            HttpGet httpget = new HttpGet(address);
            response = httpclient.execute(httpget);
            HttpEntity entity = response.getEntity();
            is = entity.getContent();
        }catch(Exception e){
            Log.e("log_tag", "Error in http connection "+e.toString());
        }
        //convert response to string
    }
}

```

```

        try{
            BufferedReader reader = new BufferedReader(new
InputStreamReader(is,"iso-8859-1"),8);
            StringBuilder sb = new StringBuilder();
            String line = null;
            while ((line = reader.readLine()) != null) {
                sb.append(line + "\n");
            }
            is.close();
            result=sb.toString();
        }catch(Exception e){

        }
        if(result.length()==5)
        {
        }
        else {
            //parse json data
            return result;
        }
        return result;
    }
    public void recebeIntervencoesApagadas(String urlPost)
    {
        String result="";

        result=SqlHttpHelper(urlPost);
        try{
            JSONArray jArray = new JSONArray(result);
            for(int i=0;i<jArray.length();i++){
                JSONObject json_data = jArray.getJSONObject(i);
                String query= "DELETE FROM intervencoes where
idIntervencao='"+json_data.getString("num_intervencao")+"' and
inativo='0'";
                BD.execSQL(query);

            }
        }
        catch(Exception erro){

        }

    }

    public void recebeIntervencoes(String urlPost)
    {
        String result="";
        result=SqlHttpHelper(urlPost);
        try{
            JSONArray jArray = new JSONArray(result);
            for(int i=0;i<jArray.length();i++){
                JSONObject json_data = jArray.getJSONObject(i);
                String query= "insert or replace into intervencoes
(idIntervencao,data_intervencao,idArvore,Motivo,idProdutor,quantidade,
observacao,estado, inativo)
values('"+json_data.getString("num_intervencao")+"','"+json_data.getSt
ring("Data_Intervencao")+"','"+json_data.getString("Arvores_idArvores"
)+"','"+json_data.getString("Motivo")+"','"+json_data.getString("Arvor
es_Produtor_idProdutor")+"','"+json_data.getString("Quantidade")+"','"

```

```

+json_data.getString("observacao")+"'','1','"+json_data.getString("inat
ivo")+"'");
        BD.execSQL(query);
        Log.i("ANTES DE APAGAR","");
    }
}
catch(Exception erro){

}

}

public void recebeIntervencoesAtualizadas(String urlPost)
{
    String result="";
    result=SqlHttpHelper(urlPost);
    try{
        JSONArray jsonArray = new JSONArray(result);
        for(int i=0;i<jArray.length();i++){
            JSONObject json_data = jsonArray.getJSONObject(i);
            String query= "update intervencoes set
idIntervencao='"+json_data.getString("num_intervencao")+"',data_interv
encao='"+json_data.getString("Data_Intervencao")+"',idArvore='"+json_d
ata.getString("Arvores_idArvores")+"',Motivo='"+json_data.getString("M
otivo")+"',idProdutor='"+json_data.getString("Arvores_Produtor_idProdu
tor")+"',quantidade='"+json_data.getString("Quantidade")+"',observacao
='"+json_data.getString("observacao")+"',estado='1',
inativo='"+json_data.getString("inativo")+"'' where
idIntervencao='"+json_data.getString("num_intervencao")+"'' and
estado='1'";
            BD.execSQL(query);

        }
    }
    catch(Exception erro){

}

}

public void chamaConsultaInterTodas()
{
    if (ConsultaArvoreIntervencaoTodas())
    {
        listener();
        mainLayout.setVisibility(LinearLayout.GONE);
        intervencaoLayout.setVisibility(LinearLayout.VISIBLE);
        mostrarDadosInter();
    }
    else
    {
        msg("Aviso!", "Não possui registros gravados");
        chamaConsulta();
        inicializaobjetos();
        listener();
        return;
    }
}
}

```

```

    public boolean ConsultaArvoreIntervencaoTodas() {
        try {
            cursor_consulta = BD.query("intervencoes",
                new String []
            {"idIntervencao","data_intervencao","idArvore","Motivo","idProdutor","
            quantidade","observacao","estado"},
                "idProdutor like '"+id.toString()+"' and inativo
            like '0'",//"estado like '%0%'",//selection,null,//"estado like
            '%0%'",//selection,
                null,//selectionArgs,
                null,//groupBy,
                null,//having,
                "idArvore",//"order by nome"//orderBy)
                null); // Limite de registros retornados
            id_arvoreCons=cursor_consulta.getColumnIndex("idArvore");

            idInterCons=cursor_consulta.getColumnIndex("idIntervencao");

            data_scanner=cursor_consulta.getColumnIndex("data_intervencao");
            motivoInter=cursor_consulta.getColumnIndex("Motivo");
            quantidade=cursor_consulta.getColumnIndex("quantidade");
            estadoConsInter=cursor_consulta.getColumnIndex("estado");

            idProdutorConsInter=cursor_consulta.getColumnIndex("idProdutor");
            obsCons=cursor_consulta.getColumnIndex("observacao");
            int numeroRegistros = cursor_consulta.getCount();
            if (numeroRegistros != 0)
            {
                // no java puro resultsewt.first();
                cursor_consulta.moveToFirst(); //posiciona no primeiro
registro
                return true;
            }
            else
                return false;

        }
        catch(Exception erro) {
            msg("Erro Banco", "Erro buscar dados no banco:
            "+erro.getMessage());
            return false;
        }
    }

    public boolean ConsultaArvoreIntervencao()
    {
        try {
            cursor_consulta = BD.query("intervencoes",
                new String []
            {"idIntervencao","data_intervencao","idArvore","Motivo","idProdutor","
            quantidade","observacao","estado"},
                "idProdutor like '"+id.toString()+"' and
            idArvore='"+edIdInter.getText().toString()+"' and inativo like
            '0'",//"estado like '%0%'",//selection,null,//"estado like
            '%0%'",//selection,
                null,//selectionArgs,
                null,//groupBy,
                null,//having,
                null,//"order by nome"//orderBy)
                null); // Limite de registros retornados

```

```

        id_arvoreCons=cursor_consulta.getColumnIndex("idArvore");

idInterCons=cursor_consulta.getColumnIndex("idIntervencao");

data_scanner=cursor_consulta.getColumnIndex("data_intervencao");
motivoInter=cursor_consulta.getColumnIndex("Motivo");
quantidade=cursor_consulta.getColumnIndex("quantidade");
estadoConsInter=cursor_consulta.getColumnIndex("estado");

idProdutorConsInter=cursor_consulta.getColumnIndex("idProdutor");
obsCons=cursor_consulta.getColumnIndex("observacao");
int numeroRegistros = cursor_consulta.getCount();
if (numeroRegistros != 0)
{
    // no java puro resultsewt.first();
    cursor_consulta.moveToFirst(); //posiciona no primeiro
registro
    return true;
}
else
    return false;

}
catch(Exception erro) {
    msg("Erro Banco", "Erro buscar dados no banco:
"+erro.getMessage());
    return false;
}
}

public boolean ConsultaArvoreMotivo(String tipo)
{
    try {
        cursor_consulta = BD.query("intervencoes",
            new String []
{"idIntervencao","data_intervencao","idArvore","Motivo","idProdutor","
quantidade","observacao","estado"},
            "idProdutor like '"+id.toString()+"' and
Motivo='"+tipo.toString()+"' and inativo like '0'",// "estado like
'%0%'",//selection,null, //"estado like '%0%'",//selection,
            null, //selectionArgs,
            null, //groupBy,
            null, //having,
            null, // "order by nome" //orderBy)
            null); // Limite de registros retornados
        id_arvoreCons=cursor_consulta.getColumnIndex("idArvore");

idInterCons=cursor_consulta.getColumnIndex("idIntervencao");

data_scanner=cursor_consulta.getColumnIndex("data_intervencao");
motivoInter=cursor_consulta.getColumnIndex("Motivo");
quantidade=cursor_consulta.getColumnIndex("quantidade");
estadoConsInter=cursor_consulta.getColumnIndex("estado");

idProdutorConsInter=cursor_consulta.getColumnIndex("idProdutor");
obsCons=cursor_consulta.getColumnIndex("observacao");
int numeroRegistros = cursor_consulta.getCount();
if (numeroRegistros != 0)
{
    // no java puro resultsewt.first();

```

```

        cursor_consulta.moveToFirst(); //posiciona no primeiro
registro
        return true;
    }
    else
        return false;

    }
    catch(Exception erro) {
        msg("Erro Banco", "Erro buscar dados no banco:
"+erro.getMessage());
        return false;
    }
}

public void chamaConsultaMotivo(String tipo)
{

    if (ConsultaArvoreMotivo(tipo))
    {
        listener();
        mostrarDadosInterMotivo();

    }
    else
    {
        msg("Aviso!", "Nã possui registros gravados");
        chamaConsultaMotivoTodos();

        listener();
        return;
    }
}

public boolean ConsultaArvoreMotivoTodos()
{
    try {
        cursor_consulta = BD.query("intervencoes",
            new String []
{"idIntervencao","data_intervencao","idArvore","Motivo","idProdutor","
quantidade","observacao","estado"},
            "idProdutor like '"+id.toString()+"' and inativo
like '0'",//"estado like '%0%'",//selection,null, //"estado like
'%0%'",//selection,
            null, //selectionArgs,
            null, //groupBy,
            null, //having,
            "idArvore", //"order by nome"//orderBy)
            null); // Limite de registros retornados
        id_arvoreCons=cursor_consulta.getColumnIndex("idArvore");

        idInterCons=cursor_consulta.getColumnIndex("idIntervencao");

        data_scanner=cursor_consulta.getColumnIndex("data_intervencao");
        motivoInter=cursor_consulta.getColumnIndex("Motivo");
        quantidade=cursor_consulta.getColumnIndex("quantidade");
        estadoConsInter=cursor_consulta.getColumnIndex("estado");

        idProdutorConsInter=cursor_consulta.getColumnIndex("idProdutor");
        obsCons=cursor_consulta.getColumnIndex("observacao");
    }
}

```

```

        int numeroRegistros = cursor_consulta.getCount();
        if (numeroRegistros != 0)
        {
            // no java puro resultsetw.first();
            cursor_consulta.moveToFirst(); //posiciona no primeiro
registro
            return true;
        }
        else
            return false;
    }
    catch(Exception erro) {

        return false;
    }
}
public void chamaConsultaMotivoTodos()
{

    if (ConsultaArvoreMotivoTodos())
    {
        listener();
        mostrarDadosInterMotivo();
    }
    else
    {
        msg("Aviso!", "Nó possui registros gravados");
        chamaConsulta();
        inicializaobjectos();
        listener();
        return;
    }
}

public void mostrarDadosInterMotivo() {

    txtArvConsMotivo.setText("Id da Áore:
"+cursor_consulta.getString(id_arvoreCons));
    txtIdConsInterMotivo.setText("Id da intervençã:
"+cursor_consulta.getString(idInterCons));
    txtDataInterMotivo.setText("Data da intervençã:
"+cursor_consulta.getString(data_scanner));
    txtMotivoInterConsMotivo.setText("Motivo da intervençã:
"+cursor_consulta.getString(motivoInter));
    txtQuantConsMotivo.setText("Quantidade recolhida (kg):
"+cursor_consulta.getString(quantidade));

    txtEstadoConsInterMotivo.setText("Estado:"+cursor_consulta.getString(e
stadoConsInter));
    txtIdProdConsMotivo.setText("id
Produtor:"+cursor_consulta.getString(idProdutorConsInter));
    txtObsMotivo.setText("Observaçã:
"+cursor_consulta.getString(obsCons));
    btRegAntMotivo.setVisibility(Button.VISIBLE);
    btProxRegMotivo.setVisibility(Button.VISIBLE);
    if(cursor_consulta.getString(quantidade).compareTo("")==0)
    {
        txtQuantConsMotivo.setVisibility(TextView.GONE);
    }
    else

```



```

        {
            txtQuantConsMotivo.setVisibility(View.VISIBLE);
        }
    }

    public void chamaConsultaMotivoPorId(String tipo)
    {
        if (ConsultaArvoreMotivoPorId(tipo))
        {
            listener();
            mostrarDadosInterMotivo();
        }
        else
        {
            msg("Aviso!", "Não possui registros gravados");
            chamaConsultaMotivoTodos();
            listener();
            return;
        }
    }

    public boolean ConsultaArvoreMotivoPorId(String tipo)
    {
        try {
            cursor_consulta = BD.query("intervencoes",
                new String []
            {"idIntervencao","data_intervencao","idArvore","Motivo","idProdutor","
            quantidade","observacao","estado"},
                "idProdutor like '"+id.toString()+"' and
            Motivo='"+tipo.toString()+"' and
            idArvore='"+idArvoreCons.toString()+"' and inativo like '0'",//"estado
            like '%0%'",//selection,null, //"estado like '%0%'",//selection,
                null, //selectionArgs,
                null, //groupBy,
                null, //having,
                null, //"order by nome"//orderBy)
                null); // Limite de registros retornados
            id_arvoreCons=cursor_consulta.getColumnIndex("idArvore");

            idInterCons=cursor_consulta.getColumnIndex("idIntervencao");

            data_scanner=cursor_consulta.getColumnIndex("data_intervencao");
            motivoInter=cursor_consulta.getColumnIndex("Motivo");
            quantidade=cursor_consulta.getColumnIndex("quantidade");
            estadoConsInter=cursor_consulta.getColumnIndex("estado");

            idProdutorConsInter=cursor_consulta.getColumnIndex("idProdutor");
            obsCons=cursor_consulta.getColumnIndex("observacao");
            int numeroRegistros = cursor_consulta.getCount();
            if (numeroRegistros != 0)
            {
                // no java puro resultsewt.first();
                cursor_consulta.moveToFirst(); //posiciona no primeiro
registro

                return true;
            }
            else
                return false;
        }
    }

```

```

    }
    catch(Exception erro) {
        msg("Erro Banco", "Erro buscar dados no banco:
+erro.getMessage());
        return false;
    }
}
public void DesactivaArvore(String arvore)
{
    String estado="0";
    String inativo="1";
    try
    {
        String sql="UPDATE arvore SET
inativo='"+inativo.toString()+"',estado='"+estado.toString()+"' Where
IDArvore='"+arvore.toString()+"' and inativo='0'";
        BD.execSQL(sql);
        String sql2="UPDATE intervencoes SET
inativo='"+inativo.toString()+"',estado='"+estado.toString()+"' Where
idArvore='"+arvore.toString()+"'";
        BD.execSQL(sql2);
        msg("Sucesso!", "Arvore apagada com sucesso!");
    }
    catch(Exception erro)
    {
    }
}
public void DesactivaIntervencao(String intervencao, String arvore)
{
    String estado="0";
    String inativo="1";
    try
    {
        String sql="UPDATE intervencoes SET
inativo='"+inativo.toString()+"',estado='"+estado.toString()+"' Where
idArvore='"+arvore.toString()+"' and inativo='0' and
idIntervencao='"+intervencao.toString()+"'";
        BD.execSQL(sql);
        msg("Sucesso!", "Intervencao apagada com sucesso!");
    }
    catch(Exception erro)
    {
    }
}
}

```

Anexo 23 – CustomHttpClient.java

```
package pt.com.android;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.URI;
import java.util.ArrayList;
import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.conn.params.ConnManagerParams;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.params.HttpConnectionParams;
import org.apache.http.params.HttpParams;

public class CustomHttpClient {
    public static final int HTTP_TIMEOUT = 30 * 10000; //
    milliseconds
    private static HttpClient mHttpClient;
    private static HttpClient getHttpClient() {
        if (mHttpClient == null) {
            mHttpClient = new DefaultHttpClient();
            final HttpParams params =
mHttpClient.getParams();

            HttpConnectionParams.setConnectionTimeout(params, HTTP_TIMEOUT);
            HttpConnectionParams.setSoTimeout(params,
HTTP_TIMEOUT);
            ConnManagerParams.setTimeout(params,
HTTP_TIMEOUT);
        }
        return mHttpClient;
    }

    public static String executeHttpPost(String url, ArrayList
postParameters) throws Exception {
        BufferedReader in = null;
        try {
            HttpClient client = getHttpClient();
            HttpPost request = new HttpPost(url);
            UrlEncodedFormEntity formEntity = new
UrlEncodedFormEntity(postParameters);
            request.setEntity(formEntity);
            HttpResponse response =
client.execute(request);
            in = new BufferedReader(new
InputStreamReader(response.getEntity().getContent()));
            StringBuffer sb = new StringBuffer("");
            String line = "";
            String NL =
System.getProperty("line.separator");
            while ((line = in.readLine()) != null) {
                sb.append(line + NL);
            }
            in.close();
            String result = sb.toString();
        }
    }
}
```

```

        return result;
    }
    finally {
        if (in != null) {
            try {
                in.close();
            }
            catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

public static String executeHttpGet(String url) throws
Exception {
    BufferedReader in = null;
    try {
        HttpClient client = getHttpClient();
        HttpGet request = new HttpGet();
        request.setURI(new URI(url));
        HttpResponse response =
client.execute(request);
        in = new BufferedReader(new
InputStreamReader(response.getEntity().getContent()));
        StringBuffer sb = new StringBuffer("");
        String line = "";
        String NL =
System.getProperty("line.separator");
        while ((line = in.readLine()) != null) {
            sb.append(line + NL);
        }
        in.close();
        String result = sb.toString();
        return result;
    }
    finally {
        if (in != null) {
            try {
                in.close();
            }
            catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
}

```

Anexo 24 – RestClient.java

```
package pt.com.android;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.List;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.ResponseHandler;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.BasicResponseHandler;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.params.BasicHttpParams;
import org.apache.http.params.HttpParams;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.ContentProviderOperation;
import android.content.ContentResolver;
import android.content.Context;
import android.database.Cursor;
import android.graphics.Typeface;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.provider.ContactsContract;
import android.provider.ContactsContract.CommonDataKinds.Phone;
import android.text.Html;
import android.text.method.LinkMovementMethod;
import android.util.Log;
import android.view.Display;
import android.view.Gravity;
import android.view.View;
import android.view.WindowManager.LayoutParams;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;
import android.widget.Toast;

public class RestClient {

    public static String convertStreamToString(InputStream is) {
        /*
         * To convert the InputStream to String we use the
         * BufferedReader.readLine()

```

```

        * method. We iterate until the BufferedReader return null which
means
        * there's no more data to read. Each line will appended to a
StringBuilder
        * and returned as String.
        */
        BufferedReader reader = new BufferedReader(new
InputStreamReader(is));
        StringBuilder sb = new StringBuilder();

        String line = null;
        try {
            while ((line = reader.readLine()) != null) {
                sb.append(line + "\n");
            }
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            try {
                is.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
        return sb.toString();
    }

    /* This is a test function which will connects to a given
    * rest service and prints it's response to Android Log with
    * labels "Praeda".
    */
}

```