

TERMO DE ABERTURA DO PROJETO

Identificação do Projeto

Projeto <i>Hortomatic – Sistema de gerenciamento de horta</i>
Unidade Demandante <i>Centro Federal de Educação Tecnológica Celso Suckow da Fonseca</i>
Gestor do Projeto <i>Daniel Romanini Garcia</i>
Patrocinador <i>Daniel Romanini Garcia, Rodolpho Vianna Santoro, Rafael Meireles e Mario Felipe</i>

1. Justificativa

O Hortomatic é um sistema para gerenciar pequenas hortas, ou seja, verificar a integridade e produtividade de um determinado tipo de plantação. A importância do projeto está, primeiramente, em poder saber como está a sua horta, onde quer que o proprietário esteja. Outro fator importante é um menor gasto de manutenção, uma vez que não haverá a necessidade de funcionários para cuidar da horta. Os principais motivos para a implantação desse sistema são: a falta de tempo, ou de funcionários para cuidar da sua pequena horta e a queda de produtividade da horta na falta de cuidadores.

2. Objetivo do Projeto

O grupo pretende colocar em prática um sistema de irrigação com Arduino integrado ao Raspberry Pi, além de sensores de captação de umidade do solo, ar e temperatura. Pretendemos utilizar uma interface em Android para facilitar o uso do proprietário, assim como um pequeno banco de dados, reunindo diferentes tipos de hortas e suas características. Um sistema de streaming poderá ser desenvolvido no decorrer do projeto e automatização das instalação de múltiplas bombas hidráulicas , se o tempo nos permitir.

3. Responsabilidades e Partes Interessadas

Unidades administrativas envolvidas: *Centro Federal de Educação Tecnológica Celso Suckow da Fonseca – Unidade Nova Iguaçu*

4. Escopo

Esperamos facilitar o gerenciamento de uma horta, assim como a interação que o dono terá com a mesma. Diminuir gastos exclusivos de monitoramentos comuns já existentes.

5. Não-Escopo

Não atenderemos o registro de informações sobre as diferentes hortas no banco de dados do aplicativo.

6. Premissas

As premissas desse projeto são o sistema de irrigação da horta e o uso de sensores que facilitam a integridade da horta.

7. Restrições

Falta de conhecimento necessário com a plataforma Android. Problema ao receber informações do servidor.

8. Projetos Inter-relacionados

Projetos de maior porte na área de agricultura

9. Riscos Iniciais

Os riscos iniciais existentes podem ser: queima de placas(arduíno ou raspberry pi), erro de conexões de jumpers, sensores queimarem, faltar carga energética para o trabalho e a planta escolhida morrer. Trocar a plataforma android pelo xamarin.

10. Tempo Estimado

De Março/2016 até Outubro/2016

11. Custo Estimado

O custo estimado está entre 200-300 reais

12. Alterações no documento

Data: 14/04/16 – Adicionado “Alterações no documento” ao documento.

Data: 05/05/16 – Alteração do “Custo Estimado”.

Data: 12/05/16 – Alteração das “Restrições”.

Data: 18/05/16 – Alteração dos “Riscos Iniciais”.


Data 10/09/16 – Compra de sensores novos

Data 29/09/16 – Fim do Projeto

13. Aprovação do Termo de Abertura

Unidade Demandante	Data	Assinatura
CEFET UneD NI	08/03/16	
Unidades Envolvidas	Data	Assinatura
Curso Téc. de Informática	08/03/16	

PRIMEIRO ARQUIVO GANTT PROJECT:



Nome	Data inicial	Data final
☐ • Integrar Arduino, Raspberry e interface web	22/03/16	07/04/16
• Estabelecer uma conexão remota entre um computador e o rasp...	22/03/16	22/03/16
• Estabelecer a conexão entre o arduino e o raspberry	23/03/16	23/03/16
• Estabelecer conexão entre o tomcat e o tornado	24/03/16	07/04/16
• Integrar o raspberry com o android	22/03/16	22/03/16
☐ • Criar as interfaces	11/04/16	13/05/16
• Criar para Python	11/04/16	13/05/16
• Criar para Android	11/04/16	13/05/16
☐ • Verificar a lista de materiais	16/05/16	16/05/16
• Verificar a necessidade e disponibilidade dos componentes	16/05/16	16/05/16
• Comprar	16/05/16	16/05/16
☐ • Conectar bombas e sensores	27/06/16	15/07/16
• Ligar o sensor de umidade	27/06/16	01/07/16
• Ligar a bomba	04/07/16	08/07/16
• Automatizar a instalação das bombas	11/07/16	15/07/16

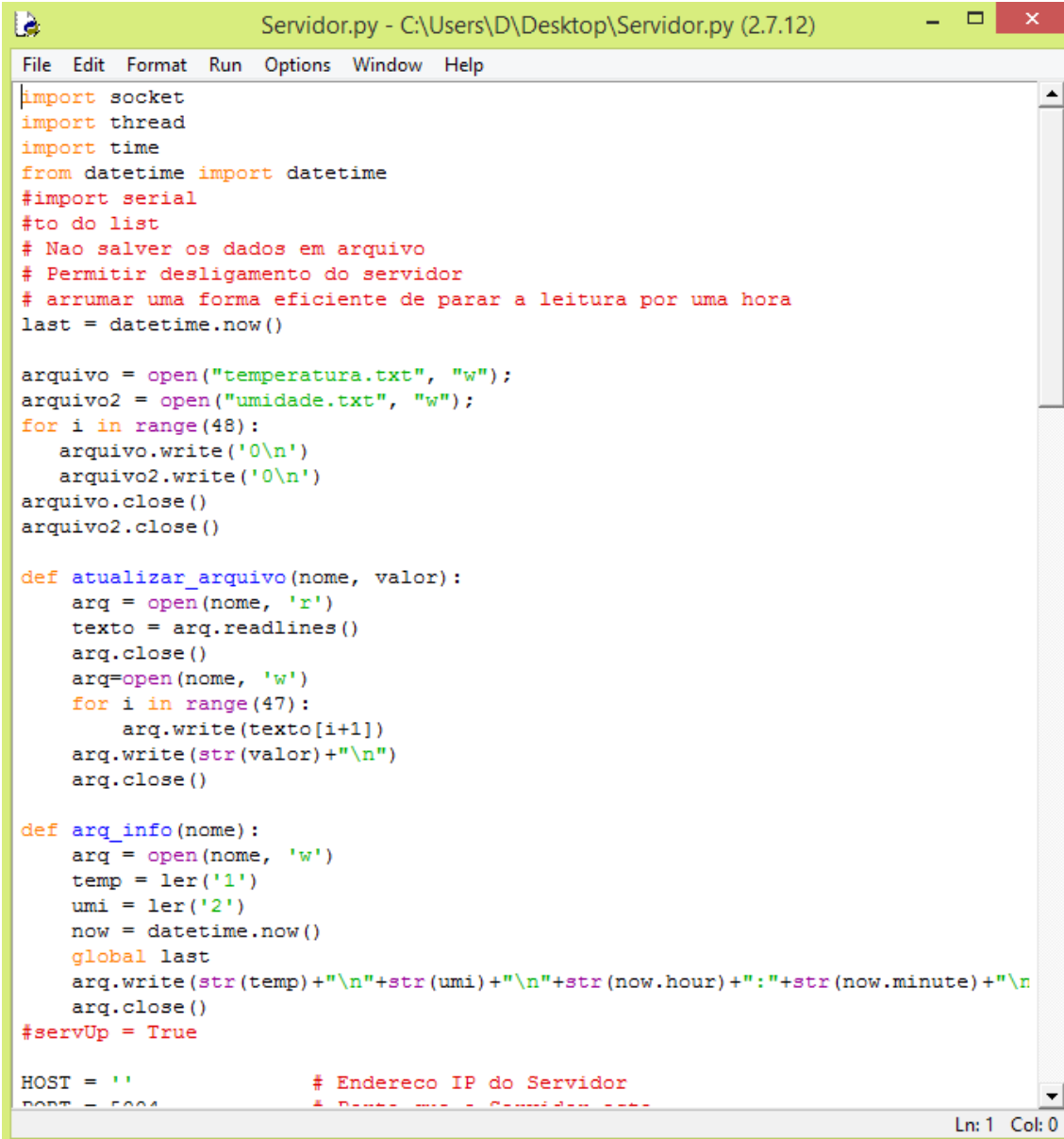
FINAL:

O projeto foi um sucesso. Acabamos gastando um pouco além do orçamento por causa do rele e do sensor que queimaram, totalizando cerca de 350 reais (aproximadamente).

A aplicação em Android que antes não estava funcionando foi concluída. O design do aplicativo mudou com relação à ideia inicial. Como resultado, acreditamos que o resultado final ficou mais amigável ao usuário.

Os sensores, por serem baratos e de procedência duvidosa, tiveram resultados baixos já esperados. Os sensores ideais utilizados em muitos projetos de irrigação teriam um custo muito alto para os nossos bolsos (Devido as taxas de importação, juros de cartão de crédito e problemas do nosso país). Porém, a meta principal foi alcançada: demonstração dos gráficos por meio de uma aplicação e adaptável a muitos contextos diferentes.

Código do Servidor:



```
Servidor.py - C:\Users\D\Desktop\Servidor.py (2.7.12)
File Edit Format Run Options Window Help
import socket
import thread
import time
from datetime import datetime
#import serial
#to do list
# Nao salvar os dados em arquivo
# Permitir desligamento do servidor
# arrumar uma forma eficiente de parar a leitura por uma hora
last = datetime.now()

arquivo = open("temperatura.txt", "w");
arquivo2 = open("umidade.txt", "w");
for i in range(48):
    arquivo.write('0\n')
    arquivo2.write('0\n')
arquivo.close()
arquivo2.close()

def atualizar_arquivo(nome, valor):
    arq = open(nome, 'r')
    texto = arq.readlines()
    arq.close()
    arq=open(nome, 'w')
    for i in range(47):
        arq.write(texto[i+1])
    arq.write(str(valor)+"\n")
    arq.close()

def arq_info(nome):
    arq = open(nome, 'w')
    temp = ler('1')
    umi = ler('2')
    now = datetime.now()
    global last
    arq.write(str(temp)+"\n"+str(umi)+"\n"+str(now.hour)+":"+str(now.minute)+"\n")
    arq.close()
#servUp = True

HOST = '' # Endereco IP do Servidor
PORT = 5004 # Porta que o Servidor esta
```

Ln: 1 Col: 0

```
Servidor.py - C:\Users\D\Desktop\Servidor.py (2.7.12)
File Edit Format Run Options Window Help
#servop = True

HOST = ''          # Endereco IP do Servidor
PORT = 5004        # Porta que o Servidor esta

def new_string(string):
    nl = []
    lista = list(string)
    for i in lista:
        if ord(i) > 31:
            nl.append(i)
    string = ''.join(nl)
    return string

def ler(valor):
    #ser = serial.Serial('/dev/ttyACM0', 9600)
    #ser.write(valor)
    #var = ser.readline()
    #ser.close()
    #return float(var);
    return float(3)

def onoff(valor):
    #ser = serial.Serial('/dev/ttyACM0', 9600)
    #ser.write(valor)
    #ser.close()
    print "oi"

def enviar_grafico(con, nome):
    arq = open(nome, 'rb')
    for i in arq.readlines():
        con.send(i)
    arq.close()

def criar_graficos():
    while True:
        atualizar_arquivo("temperatura.txt", ler('1'))
        time.sleep(1.5)
        atualizar_arquivo("umidade.txt", ler('2'))
        time.sleep(10)
```

Ln: 1 Col: 0

```
Servidor.py - C:\Users\D\Desktop\Servidor.py (2.7.12)
File Edit Format Run Options Window Help

def conectado(con, cliente):
    print 'Conectado por', cliente

    while True:
        msg = con.recv(1024)
        if not msg: break
        string = new_string(msg)
        if string == "umidade":
            print "Enviando grafico"
            enviar_grafico(con, "umidade.txt")
            break
        elif string == "temperatura":
            print "Enviando grafico"
            enviar_grafico(con, "temperatura.txt")
            break
        elif string == "Valvula ligada":
            print "Valvula ligada"
            global last
            last = datetime.now()
            onoff('3')
            con.send(msg)
            break
        elif string == "Valvula desligada":
            print "Valvula desligada"
            onoff('4')
            con.send(msg)
            break
        elif string == "info":
            arq_info(str(cliente)+".txt")
            enviar_grafico(con, str(cliente)+".txt")
            break
        else:
            con.send(msg)

    print cliente, string
    print 'Finalizando conexao do cliente', cliente
    con.close()
    thread.exit()

tcp = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

Ln: 1 Col: 0

```
tcp = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

orig = (HOST, PORT)
tcp.bind(orig)
tcp.listen(1)

thread.start_new_thread(criar_graficos, tuple([]))

while True:
    con, cliente = tcp.accept()
    thread.start_new_thread(conectado, tuple([con, cliente]))

tcp.close()
```

Ln: 1 Col: 0

Código da Aplicação (pré-final):

Cliente.java (estabelece a conexão via socket):

```
package com.example.rapha.myapplication;
import android.os.AsyncTask;
import android.widget.TextView;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.Socket;
import java.net.UnknownHostException;
public class Cliente extends AsyncTask<Void, Void, Void>{
    String address;
    int port;
    String send = "";
    String receive = "";
    TextView display;
    public Cliente(String address, int port, TextView textresponse,
String send){
        //Address 192.168.0.105
        this.address = address;
        this.port = port;
        this.display = textresponse;
        this.send = send;
    }
    @Override
    protected Void doInBackground(Void... arg0){
        Socket socket = null;
        DataInputStream in;
        DataOutputStream out;
        try{
            socket = new Socket(address, port);
            out = new DataOutputStream(socket.getOutputStream());
            in = new DataInputStream(socket.getInputStream());
            byte bytes[] = send.getBytes();
            out.writeInt(bytes.length);
            out.write(bytes);
            out.flush();
            int leng = in.readInt();
            if(leng>0){
                byte newb[] = new byte[leng];
                in.readFully(newb, 0, newb.length);
                receive = new String(newb, "US-ASCII");
            }
        } catch (UnknownHostException e){
            e.printStackTrace();
            receive = "UnknownHostException: "+e.toString();
        } catch (IOException e){
            e.printStackTrace();
            receive = "IOException: "+e.toString();
        } finally {
            if(socket != null){
                try {
                    socket.close();
                } catch (IOException e){
                    e.printStackTrace();
                }
            }
        }
        return null;
    }
    @Override
```

```

        protected void onPostExecute(Void result) {
            display.setText(receive);
            super.onPostExecute(result);
        }
    }
}

```

ActivityMain.java (Activity principal, com botões funcionais dos gráficos e válvula):

```

package com.example.rapha.myapplication;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
public class ActivityMain extends Activity{
    /** Pagina inicial, que utiliza o activity_one.xml */
    TextView response;
    EditText etip;
    EditText etport;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_one);
        etip = (EditText) findViewById(R.id.editText);
        etport = (EditText) findViewById(R.id.editText2);
        response = (TextView) findViewById(R.id.textView);
    }
    public void Solo(View v){
        String sends = "solo";
        String ip = etip.getText().toString();
        int port = Integer.parseInt(etport.getText().toString());
        Cliente cliente = new Cliente(ip, port, response, sends);
        cliente.execute();
        response.setText(sends);
        Intent intent = new Intent(getApplicationContext(),
Activity_Solo.class);
        startActivity(intent);
    }
    public void Temp(View v){
        String sends = "temp";
        String ip = etip.getText().toString();
        int port = Integer.parseInt(etport.getText().toString());
        Cliente cliente = new Cliente(ip, port, response, sends);
        cliente.execute();
        response.setText(sends);
        Intent intent = new Intent( this, Activity_Temp.class);
        startActivity(intent);
    }
    public void Valv(View v){
        String sends = "valv";
        String ip = etip.getText().toString();
        int port = Integer.parseInt(etport.getText().toString());
        Cliente cliente = new Cliente(ip, port, response, sends);
        cliente.execute();
        response.setText(sends);
        Intent intent = new Intent( this, Activity_Valv.class);
        startActivity(intent);
    }
}

```

Activity_Temp (semelhante a Activity_Solo e Activity_Valv – nessa activity está localizado o lugar onde são feitos os gráficos):

Exemplo:

```
package com.example.rapha.myapplication;
import android.app.Activity;
import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import com.androidplot.xy.LineAndPointFormatter;
import com.androidplot.xy.SimpleXYSeries;
import com.androidplot.xy.XYPlot;
import com.androidplot.xy.XYSeries;
public class Activity_Temp extends Activity {
    private XYPlot plot;
    private TextView theView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_two);
        Intent intent = getIntent();
        plot = (XYPlot) findViewById(R.id.plot);
        XYSeries s1 = new
SimpleXYSeries(SimpleXYSeries.ArrayFormat.Y_VALS_ONLY, "series1",
5,3,2,1,3,9);
        plot.addSeries(s1, new LineAndPointFormatter(Color.GREEN,
Color.GREEN, null, null));
    }
    public void myClick(View v) {
        Intent intent = new Intent( this, ActivityMain.class);
        startActivity(intent);
    }
}
```

Aplicação android baseado nos códigos:

- <http://android-er.blogspot.com.br/2014/08/bi-directional-communication-between.html>
- https://www.youtube.com/watch?v=qU1woglr_yU
- <https://www.youtube.com/watch?v=ckWG3JXCCzM>
- <https://www.youtube.com/watch?v=ViwazAAR-vE>

