

ANEXOS

A. Trabajo realizado en Fundación para la Innovación Agraria (FIA)



Figura A.1: Exterior tríptico realizado en FIA

The image shows the interior of a trifold brochure. The left panel discusses water scarcity in Chile, the middle panel highlights the device's benefits, the right panel describes the solution, and the bottom panel identifies the partner.

BENEFITS

From Copiapó to Talca south, there is a shortage of rainfall **from 20% to 60%**

Efficient irrigation is only possible with regular monitoring of soil water conditions

A SMALL CHANGE MAKES A BIG IMPACT

Make agricultural easier!

YOUR PARTNER

To Save Water

BUT WE HAVE A SOLUTION

The solution will be an intelligent, low-cost device that will efficiently determine when and how to apply water resources.

- Save water
- Get data from crops
- Make better decisions
- Remote access to information (Web and Mobile)
- Alert system for extreme weather
- keep your crops happy
- BE GREEN!**

BE GREEN!

Figura A.2: Interior tríptico realizado en FIA

Executive Summary

Smart Irrigation, saving water

Nº copy__

Our founder Nicolás Maturana, study Civil Engineering Computer at Universidad de Talca.

Water resources have been managed traditionally, before wasn't a problem because we have a lot of, but now it is, that's why we are so interested in solve this problem, if we don't have water for agriculture we will not have food.

From Copiapó to Talca south, there is a shortage of rainfall from 20% to 60%. According to CONICYT this region are the main of fruit production. The poor irrigation techniques, inadequate lining of water channels and canals, intensification of agriculture especially cropping of water intensive crops, excessive reliance on ground water and limited water recycling technologies being employed are the mains problems, but don't worry we have a solution!

The solution will be an intelligent, low-cost device that will efficiently determine, and this is very important, when and how to apply water resources to farms fields, making agriculture easier.

Benefits of using it:

- Save water
- Get data from crops
- Know how much water you use
- Make better decisions
- Remote access to information (Web and Mobile)
- Alert system for extreme weather
- Keep your crops happy
- BE GREEN!

Company Profile:
URL: www.smartirrigation.cl
Industry: Agriculture irrigation

Contact:
Nicolás Maturana
nmaturana09@gmail.com
+56-9-83020582

Financial Information:
Funding stage: Development
Capital Raised: 21 million CLP

Investors:
FIA
Family
Friends

To make it real I will need 21 million CLP. The money will be used in our team that is composed by Carlos Poblete Echeverría Ing. Agrónomo at Universidad de Talca and Dr. Benjamín Ingram Director of Department of Computer Science at Universidad de Talca.

In one year we will have the final version product, that will be easy to install, making easier for the people who want buy it.

We have one competitor who is WiseConn, but they are expensive, we can sell our product cheaper and still get profits.

The market of this product is so bigger, in Chile exists 221.915 hectares only of fruit growing (grape, cherry, citrus fruit, plum, etc.) which we hope to get 2%. So we have a big market.

The logo features a stylized blue water droplet shape above the text "Every Drop counts" in a bold, sans-serif font. A small "TM" symbol is located at the bottom right of the text.

Every Drop counts™



Figura A.3: Resumen ejecutivo

The slide consists of four panels arranged in a 2x2 grid, each featuring a green grass border at the bottom.

- Panel 1:** Contains logos for Innovate UK, Newton-Picarte Fund, and FIA (Fundación para la Innovación Agraria). It features a blue water drop icon and the text "Smart Irrigation™". Below the icons is a list:
 - Nicolás Maturana
 - The project start up at 2015
 - FIA help us
 - Three specialist people
- Panel 2:** Titled "Sector", it lists:
 - Agriculture
 - Management and control of the irrigation
- Panel 3:** Titled "Problem", it includes a map of Chile with red dots indicating specific locations. Below the map is a list:
 - Shortage of rainfall **from 20% to 60%**
 - The poor irrigation techniques
 - Intensification of agriculture especially cropping of water intensive crops
 - Limited water recycling technologies
- Panel 4:** Titled "The Customers", it lists:
 - Farmers that has irrigation systems (customer profile)
 - It can be used in home gardens
 - In Chile exists 221.915 hectares only of fruit growing (Size of the market)
 - There is only one big competitor and our device is simpler than the other one (opportunity)

1

Figura A.4: Presentación parte 1

Solution <ul style="list-style-type: none">● We offer an smart, automatic and autonomous irrigation device, making agriculture easier.● We must use water efficiently● Without water we don't have food! 	Benefits <ul style="list-style-type: none">● Save water● Get data from crops● Make better decisions● Remote accesses to information (Web and Mobile)● Keep your crops happy● Be green 
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

To make it real... <ul style="list-style-type: none">● We build a prototype on one year● 21 millions CLP● We need two specialist people, raw materials● What money are you asking for?<ul style="list-style-type: none">● Friends and family loan● FIA loan 	<p>Innovate UK Newton-Picarte Fund FIA Fundación para la Innovación Agraria</p>  <ul style="list-style-type: none">● Grateful thanks to FIA and Newton Fund for this opportunity● Thank you for listening● Any questions?  
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Figura A.5: Presentación parte 2

B. Código fuente

B.1. gps.py

Listing B.1: gps.py

```
#!/usr/bin/python
# Written by Dan Mandle http://dan.mandle.me September 2012
# License: GPL 2.0

import os
from gps import *
from time import *
from id import getserial
import time
import threading
import sys
import urllib2, urllib

gpsd = None #setting the global variable
os.system('clear') #clear the terminal (optional)

class GpsPoller(threading.Thread):
    def __init__(self):
        threading.Thread.__init__(self)
        global gpsd #bring it in scope
        gpsd = gps(mode=WATCHENABLE) #starting the stream of info
        self.current_value = None
        self.running = True #setting the thread running to true

    def run(self):
```

```

global gpsd
while gpss . running :
    gpsd . next () #this will continue to loop and grab EACH set of gpsd info to clear the buffer

if __name__ == '__main__':
    gpss = GpsPoller () # create the thread
    try:
        gpss . start () # start it up
        while True:
            #It may take a second or two to get good data
            #print gpsd.fix.latitude , ' , ', gpsd.fix.longitude , ' Time: ' , gpsd. utc
            os . system ( 'clear' )
            if gpsd . fix . latitude != 0.0 and not (math . isnan (gpsd . fix . latitude )):
                mydata=[( 'id' , getserial () ) ,( 'lat' , gpsd . fix . latitude ) ,( 'lon' ,
                    gpsd . fix . longitude )]
                mydata=urllib . urlencode (mydata)
                path='http://bri2 . utalca . cl/~nmaturana/insertCoor . php'
                req=urllib2 . Request (path , mydata)
                req . add_header ("Content-type" , "application/x-www-form-
                    urlencoded")
                page=urllib2 . urlopen (req) . read ()
                print page
                gpsd . running = False
                sys . exit (0)
                time . sleep (5) #set to whatever
            except (KeyboardInterrupt , SystemExit): #when you press ctrl+c
                print "\nKilling - Thread ... "
                gpss . running = False
                gpss . join () # wait for the thread to finish what it's doing
                print "Done.\nExiting."

```

B.2. riego.py

Listing B.2: riego.py

```

#!/usr/bin/python
import datetime
import time
from id import getserial
from clima import lluvia
from serial import Serial
import RPi.GPIO as GPIO
import urllib, json

#BCM -> maneja los pines por los numeros de GPIO17
#BOARD -> maneja los pines por número de secuencia 1, 2, 3...
GPIO.setmode(GPIO.BCM)
GPIO.cleanup()
GPIO.setwarnings(False)
GPIO.setup(27,GPIO.OUT)

#probabilidad de lluvia
#lluvia(latitud, longitud)

#obtener serial
#getserial()

serial = getserial()
url = "http://bri2.ugal.cl/~nmaturana/getCoordenada.php?id=" + serial
response = urllib.urlopen(url)
data = json.loads(response.read())
suelo = data["suelo"]

arduino = Serial('/dev/ttyUSB0', 9600, timeout=1)

while True:
    hora = datetime.datetime.now().time()
    #Se prende desde las 6 a.m. hasta las 23.59 p.m.
    if (hora >= datetime.time(6,00) and hora <= datetime.time(6,20)):
        entradas = arduino.readline()
        datos = entradas.split(" ")

```

```
if len(datos) == 2:  
    humedad = datos[0]  
    if humedad < 700:  
        if suelo == "arena":  
            #Riego arenoso  
            GPIO.output(17,GPIO.HIGH)  
            time.sleep(1200)#Riego por 20  
            min  
        else if suelo == "limoso":  
            #Riego limoso  
            GPIO.output(17,GPIO.HIGH)  
            time.sleep(1200)#Riego por 20  
            min  
    else:  
        #Riego Arcilloso  
        GPIO.output(17,GPIO.HIGH)  
        time.sleep(1200)#Riego por 20  
        min  
else:  
    GPIO.output(17,GPIO.LOW)  
    time.sleep(60)
```

B.3. clima.py

Listing B.3: clima.py

```

#!/usr/bin/python
import urllib2
import json
import time

#f = urllib2.urlopen('http://api.wunderground.com/api/e7d949410a7481dc/
#    forecast10day/lang:SP$'
#json_string = f.read()
#parsed_json = json.loads(json_string)

def lluvia(lat, lon):
    f = urllib2.urlopen('http://api.wunderground.com/api/
        e7d949410a7481dc/forecast10day/lang:SP/q/'+ str(lat) +' , '+
        str(lon) +'.json')
    json_string = f.read()
    parsed_json = json.loads(json_string)
    location = parsed_json[ 'forecast' ][ 'txt_forecast' ][ 'forecastday'
        ][0][ 'title' ]
    temp_c = parsed_json[ 'forecast' ][ 'txt_forecast' ][ 'forecastday' ,
        ][0][ 'fcttext_metric' ]
    pop = parsed_json[ 'forecast' ][ 'txt_forecast' ][ 'forecastday' ,
        ][0][ 'pop' ]
    f.close()
    return pop

```

B.4. servicio.py

Listing B.4: servicio.py

```

#!/usr/bin/python
import urllib2, urllib
from serial import Serial
import datetime
import time
import Queue
from threading import Thread

q=Queue.Queue()

def enviarBD():
    while True:
        mydata = q.get()
        mydata = urllib.urlencode(mydata)
        path = 'http://bri2.ualca.cl/~nmaturana/insertData.php'
        ,
        req = urllib2.Request(path, mydata)
        req.add_header("Content-type", "application/x-www-form-
                       urlencoded")
        page = urllib2.urlopen(req).read()
        print page

arduino = Serial('/dev/ttyUSB0', 9600, timeout=1)

t = Thread(target=enviarBD)
t.daemon=True
t.start()

while True:
    fecha = datetime.datetime.now()
    entradas = arduino.readline()
    datos = entradas.split("\u00a7")
    if len(datos) == 2:
        humedad = datos[0]
        caudal = datos[1]
        mydata=[( 'humedad' ,humedad),( 'caudal' ,caudal),( 'fecha' ,
          fecha)]
```

```
|           q.put(mydata)|
```