RFID Mini-Market Parte Web

Camila Fernandez, Fernando Socualaya y Alessia Yi

http://camilaferno.pythonanywhere.com

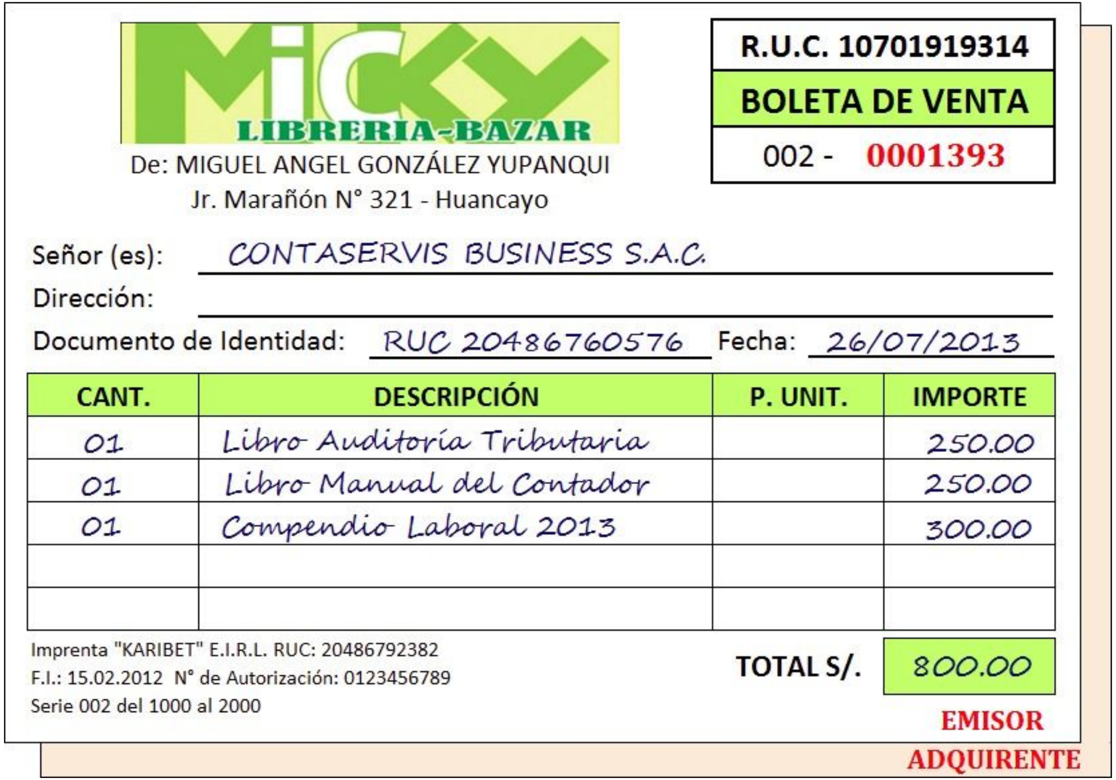
1. GITHUB

Link de Github:

https://github.com/camilaferno/MyMarket

2

La Problemática



MiCKY
LIBRERIA-BAZAR
De: MIGUEL ANGEL GONZÁLEZ YUPANQUI
Jr. Marañón N° 321 - Huancayo

R.U.C. 10701919314
BOLETA DE VENTA
002 - 0001393

Señor (es): CONTASERVIS BUSINESS S.A.C.

Dirección:

Documento de Identidad: RUC 20486760576 Fecha: 26/07/2013

CANT.	DESCRIPCIÓN	P. UNIT.	IMPORTE
01	Libro Auditoría Tributaria		250.00
01	Libro Manual del Contador		250.00
01	Compendio Laboral 2013		300.00

Imprenta "KARIBET" E.I.R.L. RUC: 20486792382
F.I.: 15.02.2012 N° de Autorización: 0123456789
Serie 002 del 1000 al 2000

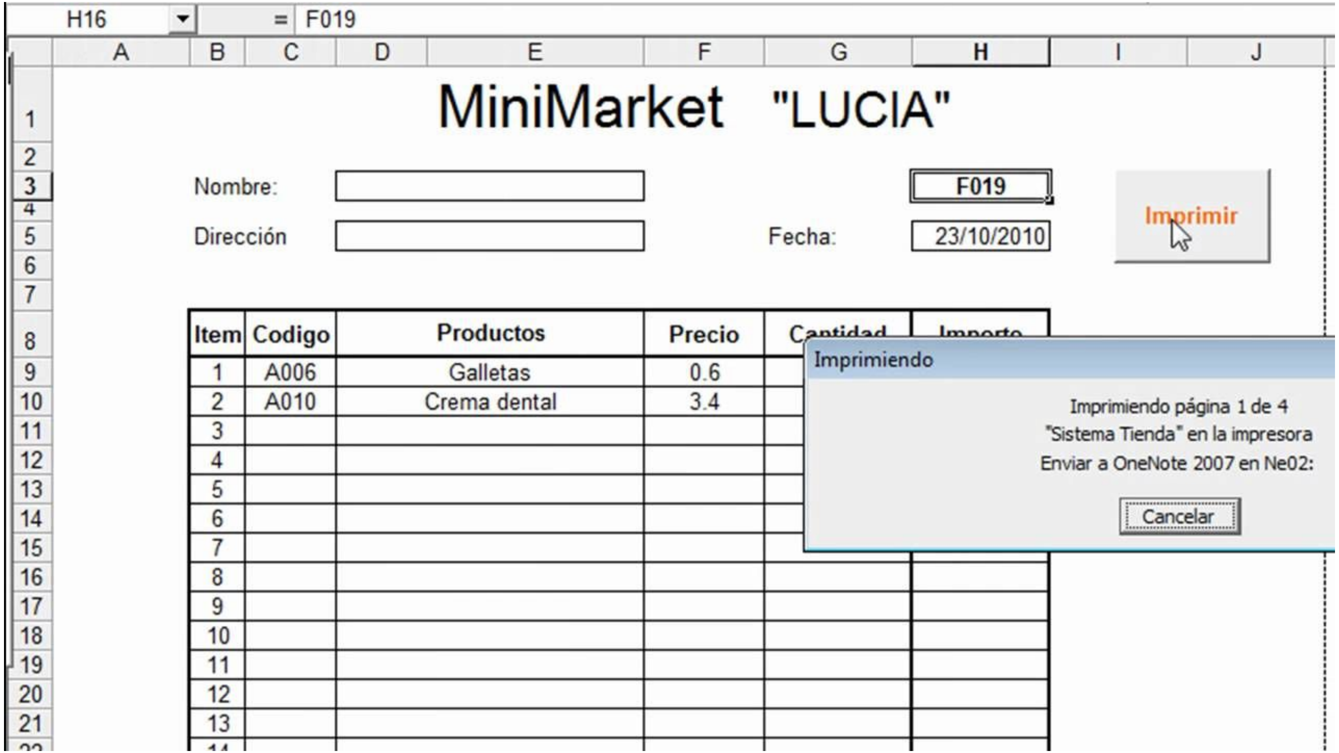
TOTAL S/. 800.00

EMISOR

ADQUIRENTE

2

La Problemática



H16 = F019

MiniMarket "LUCIA"

Nombre:

Dirección

F019

Fecha: 23/10/2010

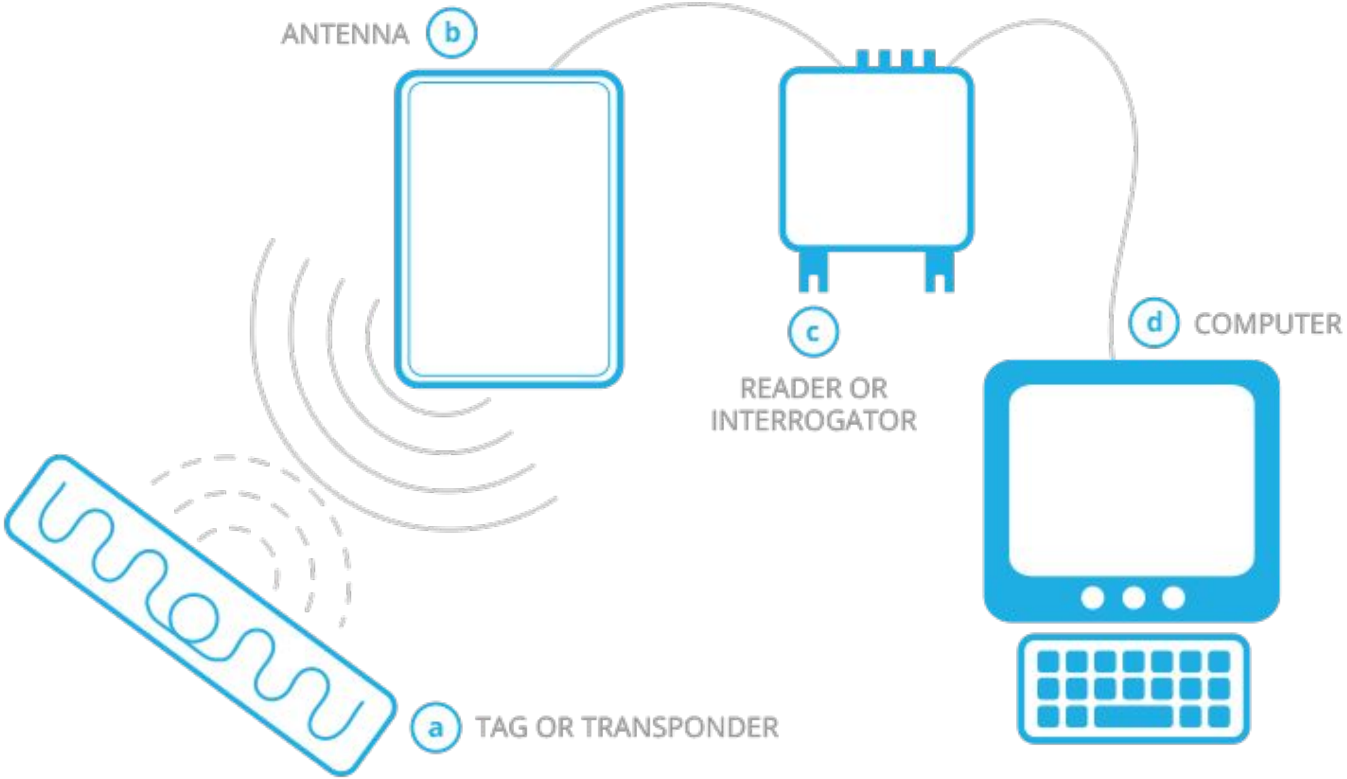
Item	Codigo	Productos	Precio	Cantidad	Importe
1	A006	Galletas	0.6		
2	A010	Crema dental	3.4		
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					

Imprimiendo

Imprimiendo página 1 de 4
"Sistema Tienda" en la impresora
Enviar a OneNote 2007 en Ne02:

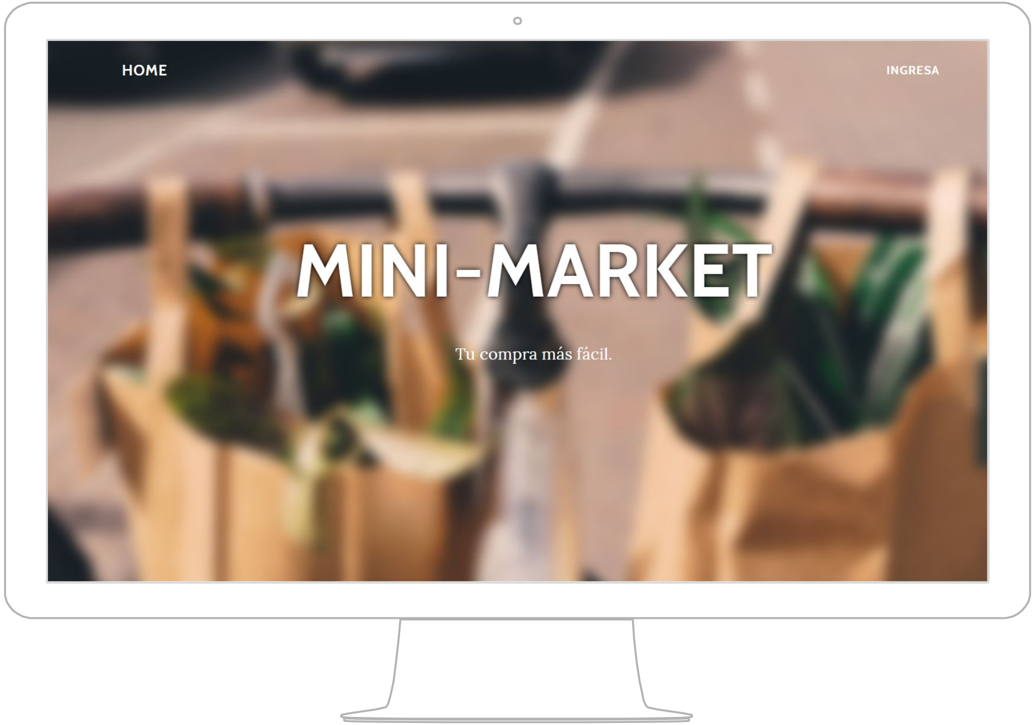
3

La Solución



4

Vista desde la computadora

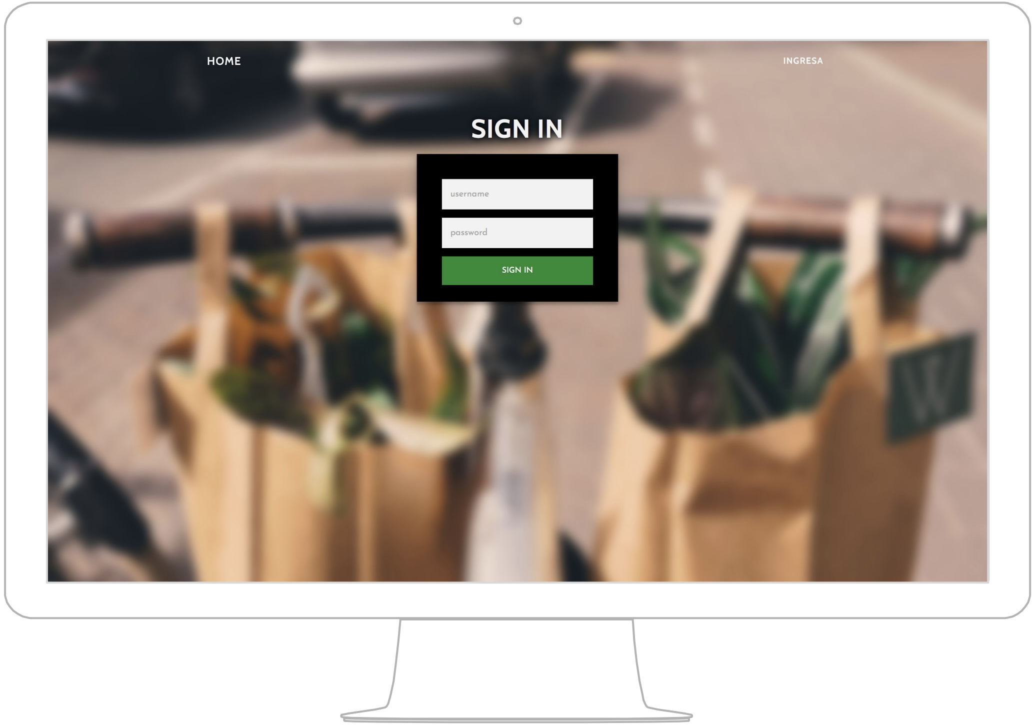


MINI-MARKET

Tu compra más fácil.

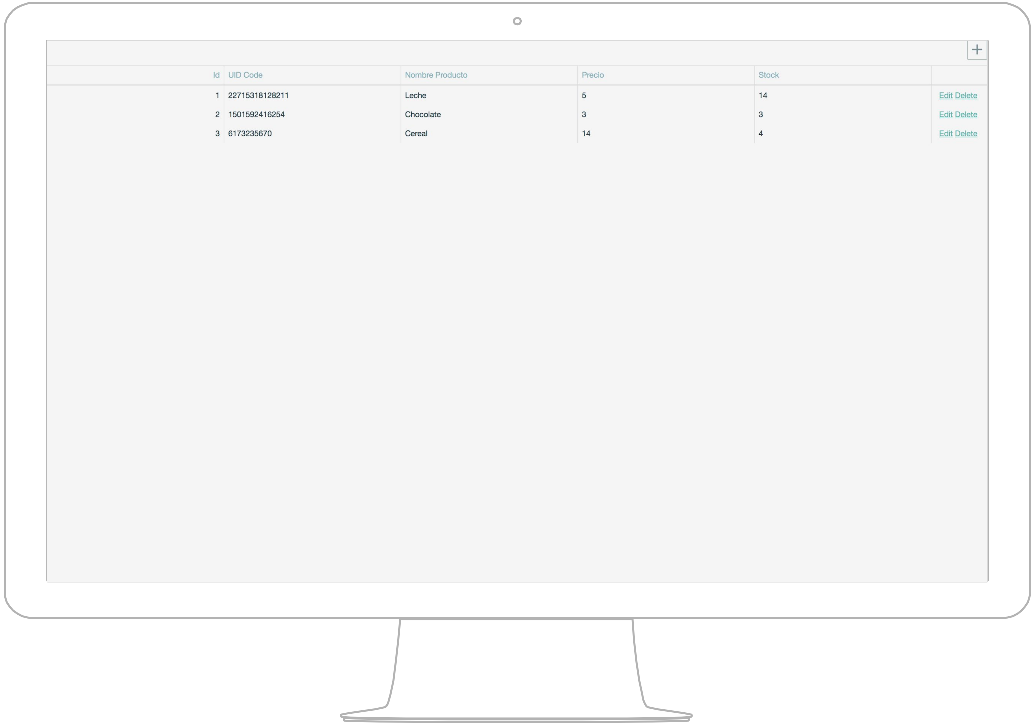
4

Vista desde la computadora



4

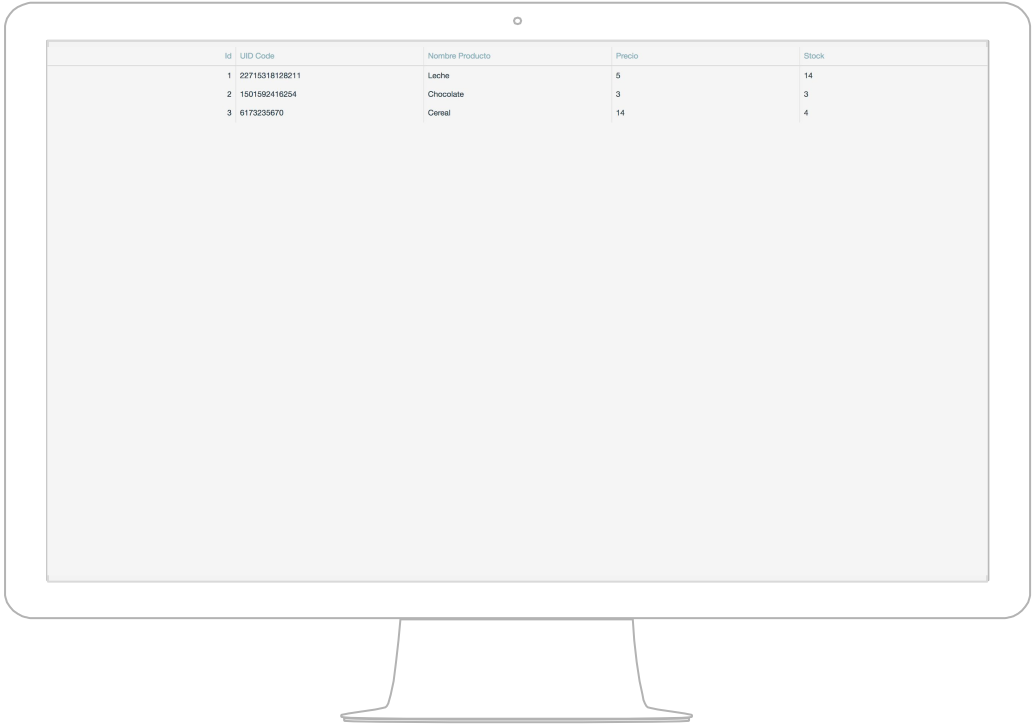
Vista desde la computadora



Id	UID Code	Nombre Producto	Precio	Stock	
1	22715318128211	Leche	5	14	Edit Delete
2	1501592416254	Chocolate	3	3	Edit Delete
3	6173235670	Cereal	14	4	Edit Delete

4

Vista desde la computadora



Id	UID Code	Nombre Producto	Precio	Stock
1	22715318128211	Leche	5	14
2	1501592416254	Chocolate	3	3
3	6173235670	Cereal	14	4

5. El CÓDIGO

El Lector

```
#   Copyright 2014,2018 Mario Gomez
<mario.gomez@teubi.co>

import RPi.GPIO as GPIO
import MFRC522
import signal
import time
import requests


###### Nuestra Base de Datos ########

Tarjeta1_id = "6173235670"
Tarjeta2_id = "22715318128211"
Tarjeta3_id = "1501592416254"

###### Nuestra Base de Datos ########

continue_reading = True

# Capture SIGINT for cleanup when the script is aborted
def end_read(signal,frame):
  global continue_reading
  print "Ctrl+C captured, ending read."
  continue_reading = False
  GPIO.cleanup()

# Hook the SIGINT
signal.signal(signal.SIGINT, end_read)
```

```
# Create an object of the class MFRC522
MIFAREReader = MFRC522.MFRC522()
# Welcome message
print "Welcome to the MFRC522 data read example"
print "Press Ctrl-C to stop."

# This loop keeps checking for chips. If one is near it will get the UID and
authenticate
while continue_reading:

  # Scan for cards
  (status,TagType) =
MIFAREReader.MFRC522_Request(MIFAREReader.PICC_REQIDL)

  # If a card is found
  if status == MIFAREReader.MI_OK:
    print "Card detected"

  # Get the UID of the card
  (status,uid) = MIFAREReader.MFRC522_Anticoll()

  # If we have the UID, continue
  if status == MIFAREReader.MI_OK:

    #Transforma el UID a string
    uid1=str(uid[0]) + str(uid[1]) + str(uid[2]) + str(uid[3]) + str(uid[4])
    print "uid1= ", uid1

# Print UID
    print "Card read UID: %s,%s,%s,%s,%s" % (uid[0], uid[1], uid[2], uid[3], uid[4])
```

```
    # This is the default key for authentication
    key = [0xFF,0xFF,0xFF,0xFF,0xFF,0xFF]

# Select the scanned tag
    MIFAREReader.MFRC522_SelectTag(uid)

    # Authenticate
    status = MIFAREReader.MFRC522_Auth(MIFAREReader.PICC_AUTHENT1A, 8,
key, uid)


    # Check if authenticated
    if status == MIFAREReader.MI_OK:
      MIFAREReader.MFRC522_Read(8)
      MIFAREReader.MFRC522_StopCrypto1()

      #Sends request to the server
      print '{"nombre": "'+uid1+'"}'

r=requests.post('http://camilaferno.pythonanywhere.com/compras','{"nombre":
"'+uid1+'"}', headers={'Content-Type':'application/json'})
      if r.status_code==200:
        print "good datos"
      else:
        print 'bad request'

    else:
      print "Authentication error"

    #Espera un rato antes de leer las tarjetas
    time.sleep(2)
```

El servidor (Base de datos)

```
class Productos(db.Model):
    __tablename__='Base de Datos'
    id = db.Column(db.Integer, primary_key=True, autoincrement=True)
    UID_Code = db.Column(db.String(50))
    Nombre_Producto = db.Column(db.String(50))
    Precio = db.Column(db.Integer)
    Stock = db.Column(db.Integer)

db.create_all()
@app.before_first_request
def set_database():
    #si la base de datos esta vacia
    if (Productos.query.all()==[]):
        db.session.add(Productos(id=1, UID_Code="22715318128211", Nombre_Producto="Leche",Precio=5, Stock=16))
        db.session.add(Productos(id=2, UID_Code="1501592416254", Nombre_Producto="Chocolate", Precio=3, Stock=4))
        db.session.add(Productos(id=3, UID_Code="6173235670", Nombre_Producto="Cereal", Precio=14, Stock=5))

        db.session.commit()
```

El servidor (Parte RFID)

```
#app route para disminuir Stock -1
@app.route("/compras", methods=["POST"])
def postcompras():
    if request.method == "POST":
        form =request.get_json()
        nombre = form['nombre']
        query=Productos.query.filter_by(UID_Code=nombre).first() #agarra la fila del UID que escaneas
        if query != None:
            if query.Stock > 1: #para que no haya un Stock negativo
                query.Stock=query.Stock-1
                db.session.commit()
                return 'ok'
            else:
                return 'no stock'
        else:
            return 'error'
    else:
        return 'error'
```

El servidor (Log-in)

```
##Codigo de CRUD para el usuario admin
##Log-in
@app.route("/")
def index():
    return render_template("index.html")

@app.route("/login")
def login():
    return render_template("login.html")

@app.route("/dashboard", methods=["POST"])
def dashboard():
    data = request.form
    if data['username'] == 'admin' and data['password'] == '1234':
        return render_template("crud_database.html")
    elif data['username'] == 'cajero' and data['password'] == '1234':
        return render_template("staticTable.html")
    else:
        return redirect("/login")
```

El Servidor (Página para cajeros)

Server.py

```
@app.route('/table')
def Page_for_cashier():

    inventario = Productos.query.all()

    database = "["
    for x in inventario:
        database += '{"id":' + str(x.id) + ', "UID_Code":" ' +
str(x.UID_Code) + '","Nombre_Producto":"' +
str(x.Nombre_Producto) + '","Precio":"' + str(x.Precio) +
'","Stock":"' + str(x.Stock) + '"},'
    database = database[0:len(database) - 1]  #borras
ultima coma
    database += ']'

    return database
```

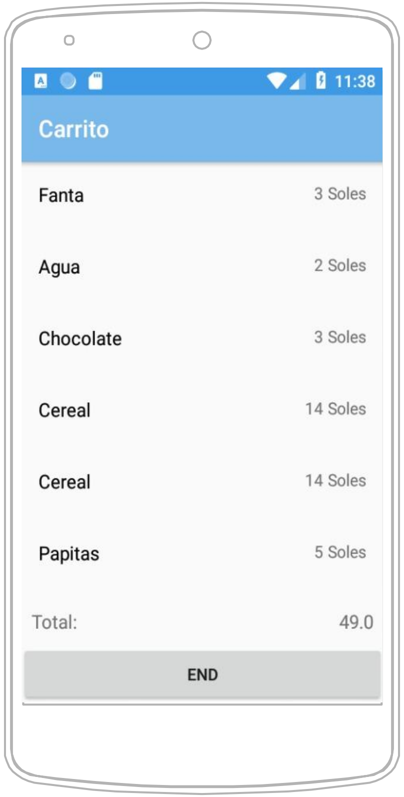
data.js

```
var var_table
$.getJSON("http://camilaferno.pythonanywhere.com/
table", function(data){
    var_table = data
})
alert("Usted no está apto para realizar cambios")
```

RFID Mini-Market
Parte Android

La problemática

Screenshot



Carrito

Fanta	3 Soles
Agua	2 Soles
Chocolate	3 Soles
Cereal	14 Soles
Cereal	14 Soles
Papitas	5 Soles

Total: 49.0