
Opinia Documentation

Release 1.0

Augusto Jair y Pedro Ludueña

19 de November de 2012

Índice general

1. Opinía	3
1.1. ¿Qué es Opinía?	3
1.2. Implementación	3
1.3. Limitaciones	3
1.4. Requisitos	3
1.5. Modo de Uso	4
1.6. Acerca De	4
1.7. Algunas Imágenes	5
1.8. Descarga	5
1.9. Código Python	5

Contenidos:

Opinia

1.1 ¿Qué es Opinia?

Opinia es un software dedicado hacia las empresas interesadas en conocer la experiencia de sus clientes, a través de un sistema de opiniones que luego se reúnen en una base de datos general.

1.2 Implementación

Para poder implementar Opinia en una compañía, solo es necesaria la descarga del software desde su repositorio online (detallado más abajo) y la disposición de un terminal para su puesta en práctica donde el cliente contrará con la posibilidad de comunicarse con la empresa. Además se debe tener en cuenta que Opinia debe ser configurado como lo indica el programa en los datos de acceso FTP y SMTP. Para ello sólo se debe modificar el Código Fuente en Python, los datos que se encuentren entre asteriscos.

1.3 Limitaciones

Las limitaciones que presenta el programa son la imposibilidad de agregar caracteres de utf-8 en los campos de Ingreso de Texto. Esto quiere decir que símbolos que no estén presentes en el inglés, como por ejemplo la ñ, la aperturas de signos (¿ o ¡), los tildes, etc.

1.4 Requisitos

Los requisitos mínimos exigidos para la implementación de Opinia, son los siguientes:

Nota:

- La disposición de Python 2.7 (o mayor) en el sistema operativo que fuere. Disponible Aquí: <http://www.python.org/download/>
 - La disposición de Pilas-engine (librería para Python). Disponible Aquí: <http://www.pilas-engine.com.ar/>
 - La disposición de una conexión a Internet (cualquier velocidad fuere).
-

1.5 Modo de Uso

El modo de uso es extremadamente simple, luego de configurar los datos requeridos sólo se debe tener en cuenta los siguientes pasos a seguir:

- Se debe hacer click sobre el campo “Nombre” y luego de que este se muestre como seleccionado, se debe borrar el contenido presente en él y posteriormente Escribir el Nombre del Cliente que opina.
- Se debe hacer click sobre el campo “Email” y luego de que este se muestre como seleccionado, se debe borrar el contenido presente en él y posteriormente Escribir el Email del Cliente que opina.
- Se debe hacer click sobre el Asunto deseado por el sujeto en calidad de opinador, y luego de recibir el aviso del programa que confirma la opción seleccionada, se puede proseguir al siguiente paso.
- En el caso de la calificación, simplemente se debe deslizar el campo y soltarlo en la calificación deseada.
- Se debe hacer click sobre el campo “Comentarios” y luego de que este se muestre como seleccionado, se debe borrar el contenido presente en él y posteriormente Escribir los Comentarios Adicionales del Cliente que opina.
- Luego de estos sencillos pasos, se debe hacer click en el botón “Enviar” y se debe aguardar a recibir la respuesta satisfactoria del software, esto puede tardar algún tiempo, dependiendo de la conexión a Internet y del estado de los servidores que reciben la Información.

Advertencia: Se debe tener en cuenta que no se pueden usar caracteres no presentes en la lengua inglesa como lo aclara la sección “Limitaciones”.

1.6 Acerca De

El software ha sido elaborado por dos alumnos del ITS Villada de 4° año C, de la especialidad Informática. Año 2012. Para más información puede contactarse con Augusto Jair augustojair_96@hotmail.com o con Pedro Ludueña pedro_ignacio_luduenaa@hotmail.com
El programa se encuentra con licencia GPL-3 de Creative Commons.

Opinia por <a xmlns:cc="http://creativecommons.org/ns#" href="https://github.com/augustojair/Opinia.git" property="cc:attributionName" rel="cc:attributionURL">Augusto Jair & Pedro Luduenaa se encuentra bajo una Licencia Creative Commons Atribución-CompartirIgual 3.0 Unported.
Basada en una obra en <a xmlns:dct="http://purl.org/dc/terms/" href="https://github.com/augustojair/Opinia.git" rel="dct:source">https://github.com/augustojair/Opinia.git.

1.7 Algunas Imágenes

Este ejemplo es el modelo que utiliza a la compañía básica del proyecto, el “Camping Las Vertientes”.

1.8 Descarga

La descarga de Opinia, se puede dar desde su repositorio en GitHub, ingresando el siguiente comando en la terminal de linux:

1.9 Código Python

1.9.1 Opinia.py

Nota: Se debe editar el archivo de configuración presente en “.Data/conf.ini”

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
```

```
3
4 """Importamos los modulos"""
5 import pilas
6 import sys
7 import Multilinea
8 import ConfigParser
9 import urllib
10
11 """Creamos y Configuramos todas las funciones"""
12
13 def cuando_hacen_click():
14     """Esta funcion es el evento que actuara cuando se haga click en el boton "Enviar" """
15
16
17     """Capturamos y Guardamos los datos del Formulario"""
18     pilas.avisar("Aguarde por favor, el formulario esta siendo enviado")
19     cfgFile_w = open(".Data/datos.ini",'w')
20     C.set('datos','nombre', txtNombre.texto)
21     C.write(cfgFile_w)
22     C.set('datos','email', txtEmail.texto)
23     C.write(cfgFile_w)
24     cfgFile_w.close()
25     guardarmensaje = open(".Data/mensaje.txt","w")
26     guardarmensaje.write(txtMensaje.texto)
27     guardarmensaje.close()
28
29     """Disparamos las funciones que enviara el email y otra que creara el archivo html, para luego se
30     enviar_email()
31     crear_html()
32
33     """Ahora se limpiaran los campos y se reiniciara el formulario"""
34     limpiar=open(".Data/datos.ini","w")
35     limpiar.write("[datos]\nnombre = \nemail = \nasunto = \n calificacion = ")
36     limpiar.close()
37     limpiar2=open(".Data/mensaje.txt","w")
38     limpiar2.write("")
39     limpiar2.close()
40     txtNombre.texto = "Ingrese su Nombre"
41     txtEmail.texto = "Ingrese su Email"
42     txtMensaje.texto = u"Ingrese sus Comentarios"
43     cuando_cambia_Calificacion(0)
44
45     """Si todo ha sido procesado correctamente, notificamos la aceptacion"""
46     pilas.avisar(u'Se envi  el formulario')
47     print u'Se envi  el formulario Correctamente'
48
49
50 def cuando_selecciona_Asunto(opcion_seleccionada):
51     """Esta funcion es el evento que actuara al seleccionar una alternativa del Campo de Opcion"""
52     pilas.avisar("Ha seleccionado la opcion: " + opcion_seleccionada)
53     cfgFile_w = open(".Data/datos.ini",'w')
54     C.set('datos','asunto', opcion_seleccionada)
55     C.write(cfgFile_w)
56     cfgFile_w.close()
57     return None
58
59 def cuando_cambia_Calificacion(valor):
60     """Esta funcion es el evento que actua cuando se cambia el valor del Campo Deslizante """
```

```

61     lblCalificacionTotal.definir_texto(str(int(valor * 10)) +u' puntos')
62     cfgFile_w = open(".Data/datos.ini",'w')
63     C.set('datos','calificacion', str(int(valor * 10)) +u' puntos')
64     C.write(cfgFile_w)
65     cfgFile_w.close()
66
67 def bajar_html():
68     """Esta funcion es la que descargara el archivo html del sitio web y luego se lo entregara a la """
69     def reporthook(*a): print a
70
71     C = ConfigParser.ConfigParser()
72     C.read(".Data/conf.ini")
73     web = C.get("SMTP","sitio-web")
74     url = web+"Opiniones/Opiniones.html"
75     file = ".Data/Opiniones.html"
76     urllib.urlretrieve(url, file, reporthook)
77     crear_html()
78
79 def crear_html():
80
81     """Esta es la funcion que crea el informe HTML para luego ser subidos a la web"""
82
83     from ftplib import FTP
84     C.read(".Data/conf.ini")
85     usuario = C.get("FTP","usuario")
86     servidor = C.get("FTP","servidor")
87     password = C.get("FTP","password")
88     ftp = FTP(servidor);
89     ftp.login(user=usuario, passwd=password)
90     ftp.cwd("/public_html/Opiniones/");
91     ftp.retrbinary("RETR Opiniones.html",open(".Data/Opiniones.html","wb").write)
92     ftp.quit()
93
94     """Cargamos los datos para crear el archivo html"""
95     C.read(".Data/datos.ini")
96     nombre = C.get("datos",'nombre')
97     email = C.get("datos",'email')
98     asunto = C.get("datos",'asunto')
99     calificacion = C.get("datos",'calificacion')
100    mensaje = open(".Data/mensaje.txt",'r')
101    mensajea = mensaje.read()
102    mensaje.close()
103
104    """Guardamos el archivo html"""
105    creararchivo = open(".Data/Opiniones.html","a")
106    creararchivo.write(str("\n<hr>Nombre: "+nombre+"<br>Email: <a href=mailto:"+email+">"+email+"</a>"))
107    creararchivo.close()
108
109    """Disparamos la funcion que subira el archivo html"""
110    subir_html()
111
112 def enviar_email():
113     """Esta es la funcion que enviara el email notificando de una nueva opinion"""
114
115     """Importamos la libreria necesaria y cargamos los datos para luego ser enviados"""
116     import smtplib
117     C = ConfigParser.ConfigParser()
118     C.read(".Data/datos.ini")

```

```
119     nombre = C.get("datos","nombre")
120     email = C.get("datos","email")
121     asunto = C.get("datos","asunto")
122     calificacion = C.get("datos","calificacion")
123     cargarmensaje = open(".Data/mensaje.txt","r")
124     mensaje=cargarmensaje.read()
125     cargarmensaje.close()
126     C.read(".Data/conf.ini")
127     web = C.get("SMTP","sitio-web")
128     email2 = C.get("SMTP","email")
129     clave = C.get("SMTP","password")
130
131     """Importamos los modulos adicionales necesarios"""
132     from email.mime.text import MIMEText
133
134     """Creamos el mensaje"""
135     cuerpodelmensaje = str("Nombre: "+nombre+"\n"+"E-mail: "+email+"\n"+"Calificacion: "+calificacion)
136     msg = MIMEText(cuerpodelmensaje)
137
138     """Conectamos con el server"""
139     msg['Subject'] = 'Usted ha recibido una nueva opinion - '+asunto
140     msg['From'] = email
141     msg['To'] = email2
142
143     """Autenticamos"""
144     mailServer = smtplib.SMTP('smtp.gmail.com',587)
145     mailServer.ehlo()
146     mailServer.starttls()
147     mailServer.ehlo()
148     mailServer.login(email2, clave)
149
150     """Enviamos"""
151     mailServer.sendmail(email2, email2, msg.as_string())
152
153     """Cerramos conexion"""
154     mailServer.close()
155
156 def subir_html():
157     """Esta es la funcion que subira el archivo html al servidor web"""
158
159     """Importamos las librerias necesarias"""
160     import ftplib
161     import os
162
163     C.read(".Data/conf.ini")
164     """ Cargamos Datos FTP"""
165     ftp_servidor = C.get("FTP","servidor")
166     ftp_usuario = C.get("FTP","usuario")
167     ftp_clave = C.get("FTP","password")
168     ftp_raiz = '/public_html/Opiniones/'
169
170     """ Cargamos Datos del fichero a subir"""
171     fichero_origen = '.Data/Opiniones.html'
172     fichero_destino = 'Opiniones.html'
173
174     """Conectamos con el servidor"""
175     try:
176         s = ftplib.FTP(ftp_servidor, ftp_usuario, ftp_clave)
```

```

177         try:
178             f = open(fichero_origen, 'r')
179             s.cwd(ftp_raiz)
180             s.storbinary('STOR ' + fichero_destino, f)
181             f.close()
182             s.quit()
183         except:
184             print "No se ha podido encontrar el fichero " + fichero_origen
185     except:
186         print "No se ha podido conectar al servidor " + ftp_servidor
187
188
189     """Creamos la estructura basica del software"""
190
191     """Inicializamos ConfigParser"""
192     C = ConfigParser.ConfigParser()
193     C.read(".Data/datos.ini")
194
195     """Iniciamos Pilas"""
196     pilas.iniciar(titulo=u"Opinión Acerca de Camping Las Vertientes - Opinia")
197     fondo = pilas.fondos.Fondo(".Data/Camping.jpg")
198     fondo.escala = 1
199     posicionV = 10
200     posicionL = -170
201     posicionR = 30
202     lblEscala = 0.75
203     escapar = 500
204     Multilinea.inicializar()
205
206     """ Creamos el Boton"""
207     boton = pilas.interfaz.Boton("Enviar")
208     boton.x,boton.y = 150,-200
209
210     """Convocamos el evento que actuara cuando se haga click en el boton"""
211     boton.conectar(cuando_hacen_click)
212
213     """Creamos y Configuramos Las Entradas de Texto"""
214     lblNombre = pilas.actores.Texto("Nombre:")
215     lblNombre.escala = lblEscala
216     lblNombre.x, lblNombre.y = posicionL, (posicionV *20)
217     txtNombre = pilas.interfaz.IngresoDeTexto(limite_de_caracteres=39, texto_inicial="Ingrese Su Nombre")
218     txtNombre.x,txtNombre.y = posicionR, (posicionV *20)
219     lblEmail = pilas.actores.Texto("E-mail:")
220     lblEmail.escala = lblEscala
221     lblEmail.x, lblEmail.y = posicionL, (posicionV *15)
222     txtEmail = pilas.interfaz.IngresoDeTexto(limite_de_caracteres=39, texto_inicial="Ingrese Su Email")
223     txtEmail.x,txtEmail.y = posicionR, (posicionV *15)
224
225     """Creamos y Configuramos el Campo de Opcion"""
226     lblAsunto = pilas.actores.Texto("Asunto:")
227     lblAsunto.escala = lblEscala
228     lblAsunto.x, lblAsunto.y = posicionL, (posicionV *10)
229     opcionesAsunto = pilas.interfaz.ListaSeleccion(['Queja', 'Sugerencia', 'Contacto General'], cuando_selecciona=
230     opcionesAsunto.x,opcionesAsunto.y = posicionR-100, (posicionV *10)
231
232     """Creamos el campo Mensaje"""
233     lblMensaje = pilas.actores.Texto("Mensaje:")
234     lblMensaje.escala = lblEscala

```

```
235 lblMensaje.x, lblMensaje.y = posicionL, (posicionV *-5)
236 txtMensaje = Multilinea.EntradaDeTexto(limite_de_caracteres=235, texto_inicial=u"Ingrese sus Comentarios")
237 txtMensaje.x, txtMensaje.y = posicionR, (posicionV *-5)
238
239 """Creamos y configuramos el Campo deslizante"""
240 lblCalificacion = pilas.actores.Texto(u"Calificación:")
241 lblCalificacion.escala = lblEscala
242 lblCalificacion.x, lblCalificacion.y = posicionL, (posicionV * 5)
243 lblCalificacionTotal = pilas.actores.Texto(u'0 puntos')
244 lblCalificacionTotal.escala = lblEscala
245 lblCalificacionTotal.x, lblCalificacionTotal.y = posicionL + 2 + lblCalificacion.ancho, (posicionV * 5)
246 Calificacion = pilas.interfaz.Deslizador()
247 Calificacion.conectar(cuando_cambia_Calificacion)
248 Calificacion.x, Calificacion.y = posicionR, posicionV*5
249
250 """Ejecutamos el Programa"""
251 pilas.ejecutar()
```

1.9.2 Multilinea.py

Nota: Este código es adicional, ya que Pilas no cuenta con la función Multilinea para una entrada de texto, por lo tanto el código fuente básico provisto por Pilas, ha sido editado por el equipo de “Opinia” y su código es el siguiente.

```
1  # -*- encoding: utf-8 -*-
2
3  import pilas
4  import re
5  from pilas.interfaz.base_interfaz import BaseInterfaz
6
7  class EntradaDeTexto(BaseInterfaz):
8
9      def __init__(self, texto_inicial="", x=0, y=0, ancho=300, limite_de_caracteres=20, icono=None, a):
10         BaseInterfaz.__init__(self, x=x, y=y)
11         self.texto = texto_inicial
12         self.cursor = ""
13         self._cargar_lienzo(ancho)
14         self.acepta_multilinea = acepta_multilinea
15
16         if icono:
17             self.icono = pilas.imagenes.cargar(icono)
18         else:
19             self.icono = None
20
21         self.imagen_caja = pilas.imagenes.cargar(".Data/mensaje.png")
22         self.centro = ("centro", "centro")
23         self._actualizar_imagen()
24         self.limite_de_caracteres = limite_de_caracteres
25         self.cualquier_caracter()
26
27         pilas.eventos.pulsa_tecla.conectar(self.cuando_pulsa_una_tecla)
28         pilas.mundo.agregar_tarea_siempre(0.40, self._actualizar_cursor)
29         self.fijo = True
30
31     def _actualizar_cursor(self):
32         if (self.tiene_el_foco):
```

```

33         if self.cursor == "":
34             self.cursor = "_"
35         else:
36             self.cursor = ""
37     else:
38         self.cursor = ""
39
40     self._actualizar_imagen()
41     return True
42
43 def cualquier_caracter(self):
44     self.caracteres_permitidos = re.compile(".*")
45
46 def solo_numeros(self):
47     self.caracteres_permitidos = re.compile("\d+")
48
49 def solo_letras(self):
50     self.caracteres_permitidos = re.compile("[a-z]+")
51
52 def cuando_pulsa_una_tecla(self, evento):
53     leermaximo=open(".Data/mensaje.txt", "r")
54     numeromaximo=int(leermaximo.read())
55     leermaximo.close()
56     if (self.tiene_el_foco and self.activo):
57         if evento.codigo == '\x08' or evento.texto == '\x08':
58             if numeromaximo == 0:
59                 print "0"
60             else:
61                 numeromaximo = numeromaximo+1
62                 escribirmaximo=open(".Data/mensaje.txt", "w")
63                 escribirmaximo.write(str(numeromaximo))
64                 escribirmaximo.close()
65                 print "1"
66                 self.texto = self.texto[:-1]
67         elif str(evento.texto) == '\r' and self.acepta_multilinea:
68             self.texto += '\n'
69             numeromaximo = 0
70             escribirmaximo=open(".Data/mensaje.txt", "w")
71             escribirmaximo.write(str(numeromaximo))
72             escribirmaximo.close()
73             print "2"
74         else:
75             if len(self.texto) < self.limite_de_caracteres:
76                 if numeromaximo>=25:
77                     self.texto += '\n'
78                     self.texto = self.texto + evento.texto
79                     numeromaximo=0
80                     escribirmaximo=open(".Data/mensaje.txt", "w")
81                     escribirmaximo.write(str(numeromaximo))
82                     escribirmaximo.close()
83                     print "3"
84                 else:
85                     self.texto = self.texto + evento.texto
86                     numeromaximo=numeromaximo+1
87                     escribirmaximo=open(".Data/mensaje.txt", "w")
88                     escribirmaximo.write(str(numeromaximo))
89                     escribirmaximo.close()
90                     print "4"

```

```
91         self._actualizar_imagen()
92
93     def _cargar_lienzo(self, ancho):
94         self.imagen = pilas.imagenes.cargar_superficie(ancho, 150)
95
96     def _actualizar_imagen(self):
97         ancho = self.imagen_caja.ancho()
98         alto = self.imagen_caja.alto()
99         self.imagen.pintar_parte_de_imagen(self.imagen_caja, 0, 0, 40, ancho, 0, 0)
100
101         if self.icono:
102             dx = 20
103             self.imagen.pintar_parte_de_imagen(self.icono, 0, 0, 40, ancho, 7, 7)
104         else:
105             dx = 0
106
107         for x in range(40, self.imagen.ancho() - 40):
108             self.imagen.pintar_parte_de_imagen(self.imagen_caja, ancho - 40, 0, 40, alto, x, 0)
109
110         self.imagen.texto(self.texto + self.cursor, 15 + dx, 20)
111     def inicializar():
112         inicializar=open(".Data/mensaje.txt","w")
113         inicializar.write(str(0))
114         inicializar.close()
```