



“Transformando cultivos,
transformando vidas”



Identificación de cultivos de uso ilícito en el departamento del Cauca.

- Preguntas relevantes
- Enunciados
- Cifras grandes



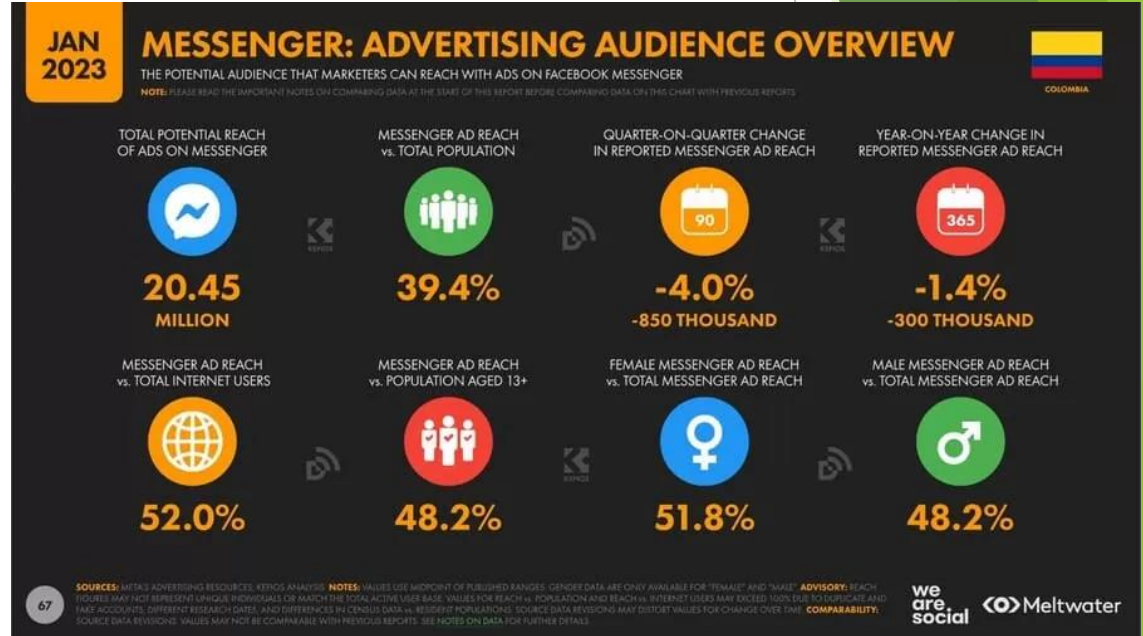


Equipo-general

- Somos un equipo de 3 personas apasionados por la tecnología, con habilidades en programación, diseño web e inteligencia artificial, que convergemos en el deseo de desarrollo y crecimiento.

Contexto del problema

- Claridad
- Cifras detalladas
- Regiones
- Personas
- Comportamientos
- Estudios





Solución : Plataforma Designious

- ▶ La solución que se plantea es intuitiva, amigable con el usuario que proporciona los datos requeridos.
- ▶ A nuestros clientes se les brinda información de su interés, clara y limpia, como también la ayuda para la comprensión y contextualización de los datos suministrados.

Solución : Por esta razón

Para el desarrollo y cumplimiento de lo anterior esta plataforma fue creado en diferentes herramientas como html,css y bibliotecas de javascript como leaflet para el diseño web, Bases de datos para el almacenamiento y Python con librerías como keras, tensor flow y opencv para el tratamiento y análisis de imágenes y datos.

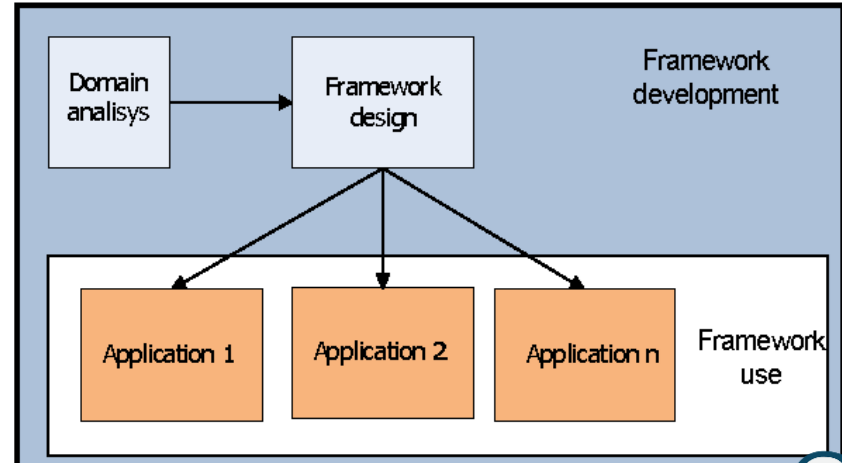
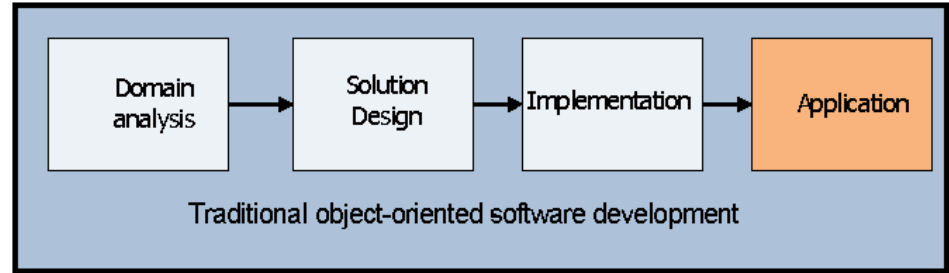


Fig. 2. Traditional object-oriented software development VS framework

```

content">
ox">
"/enviar formular
de="email" id="en
r="nota">Descripc
id="nota" name=
map"></div>
adadas selecciona
class="botoni gzn

```

```

el mapa -->
code.jquery.com/

```

```

establecer la v
ap').setView([2.

```

```

del mapa desde (
//s}.tile.oper
OpenStreetMap

```

```

hacer clic e

```

```

function (e) {
atlon.lat;
atlon.lng;

```

```

Coordenadas

```



Solución general - prototipo



Ubicacion
Municipio
Popayán

Corregimiento

DESIGNIOUS

Ubicacion
Municipio
Popayán

Vereda

Descripcion:

DESIGNIOUS

Ubicacion
Municipio
Popayán

Descripcion

enviar

Únete a la Lucha Contra los Cultivos Ilícitos con Tecnología Innovadora

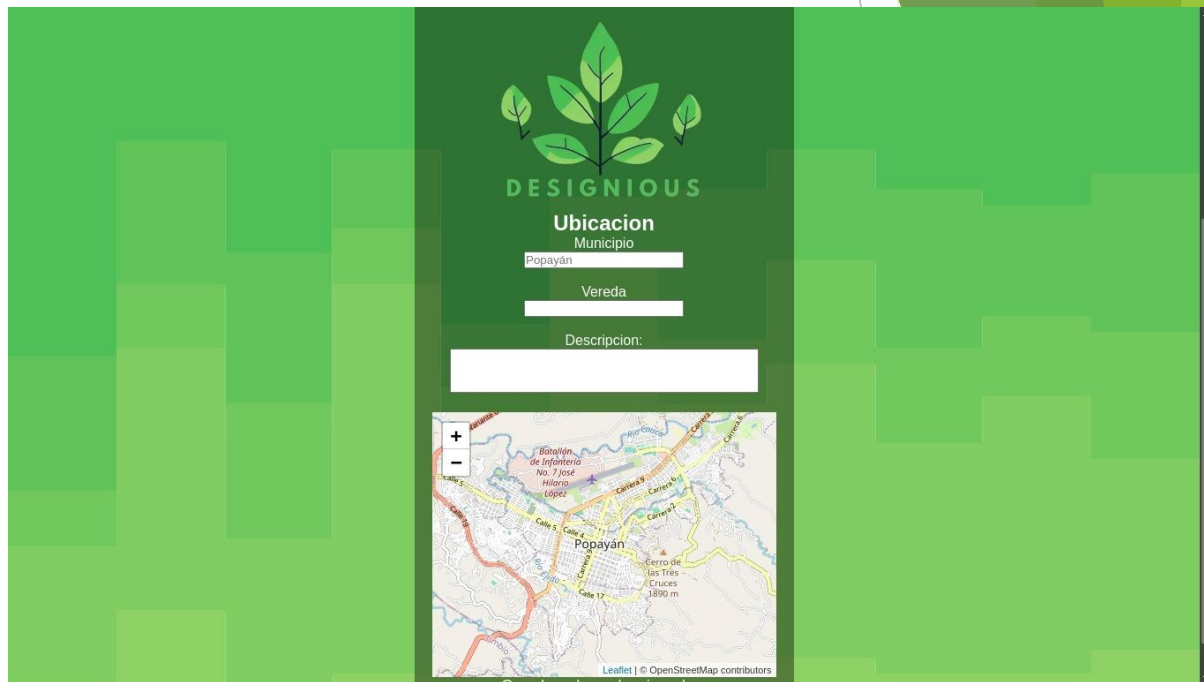
Sé parte del cambio: Reporta cultivos ilícitos de manera segura y anónima

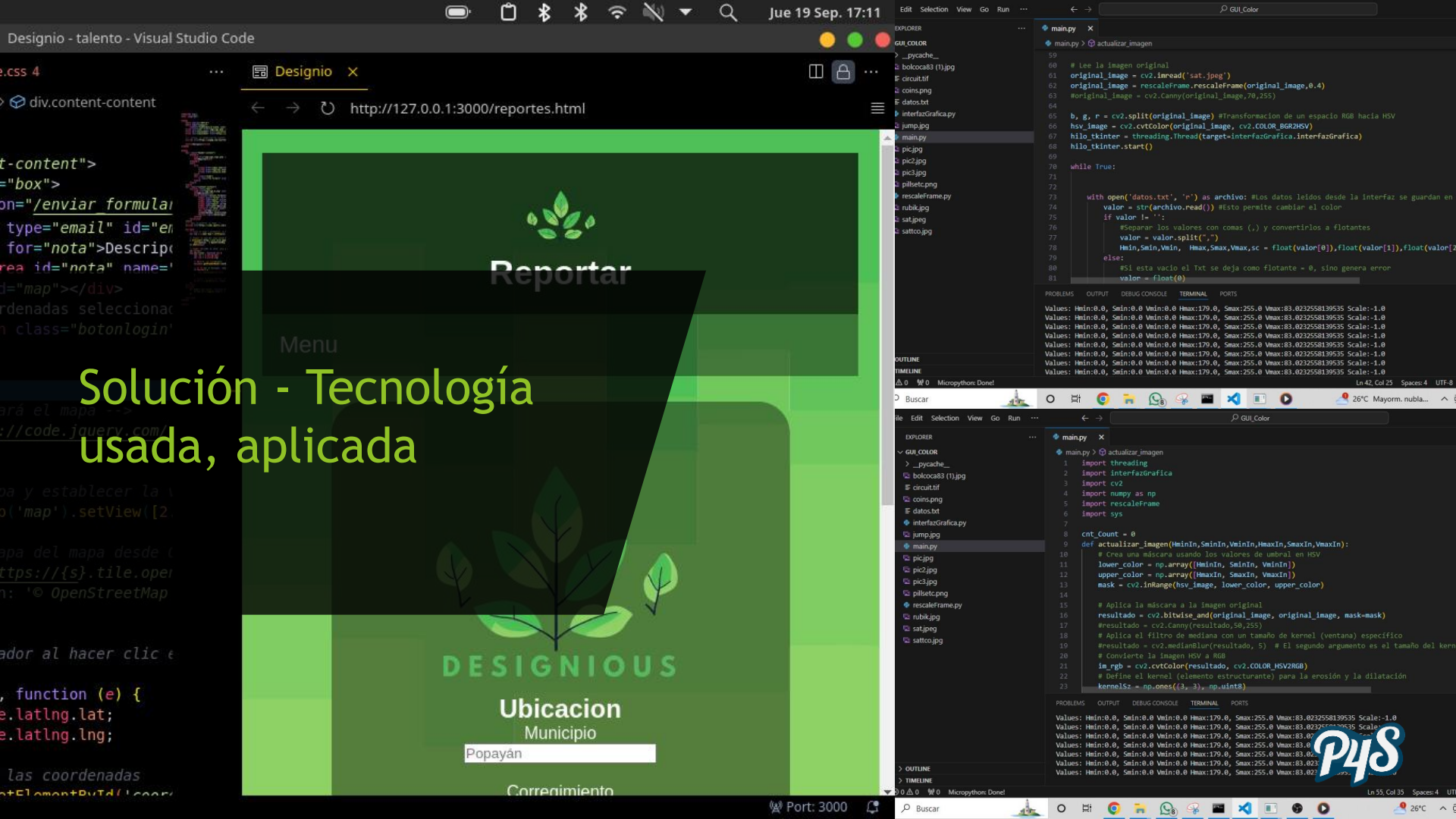
Introducción

El Departamento del Cauca enfrenta un desafío crítico con la expansión de cultivos ilícitos que afectan tanto a la comunidad como al medio ambiente. ¡Tú puedes hacer la diferencia! Utiliza herramientas tecnológicas avanzadas para reportar cultivos de uso ilícito de forma anónima y apoya a las autoridades en la lucha contra el narcotráfico.



Solución general - prototipo a detalle





Solución - Tecnología usada, aplicada

Reportar

Menu

DESIGNIOUS

Ubicacion
Municipio

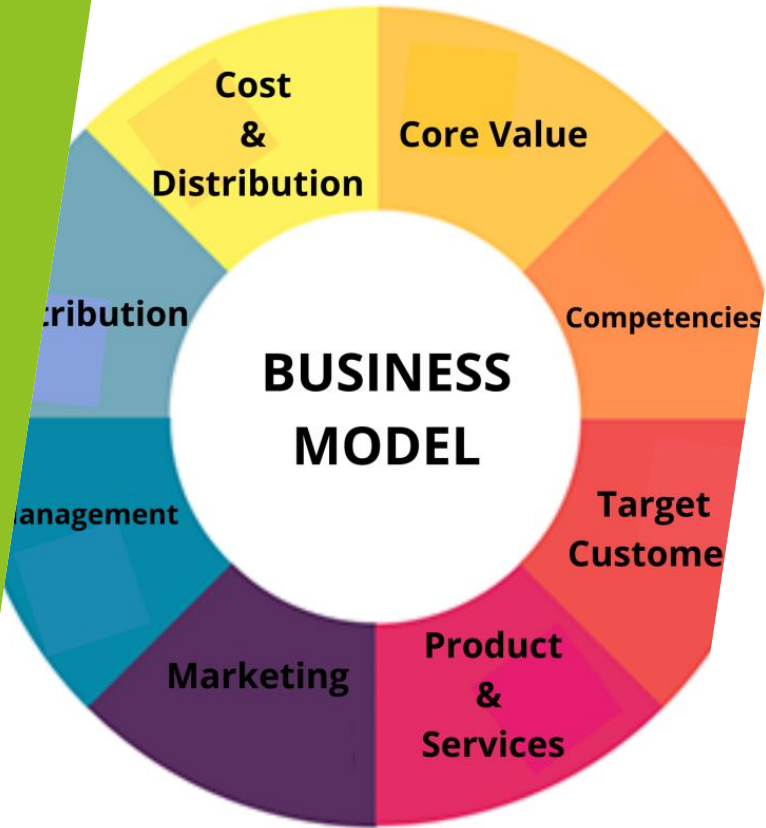
Popayán

Corredimiento

```
main.py x
main.py > actualizar_imagen
59
60 # Lee la imagen original
61 original_image = cv2.imread('sat.jpeg')
62 original_image = cv2.resizeFrame(rescaleFrame(original_image, 0.4)
63 # original_image = cv2.Canny(original_image, 70, 255)
64
65 b, g, r = cv2.split(original_image) #Transformacion de un espacio RGB hacia HSV
66 hsv_image = cv2.cvtColor(original_image, cv2.COLOR_BGR2HSV)
67 hilo_tkinter = threading.Thread(target=interfazGrafica.interfazGrafica)
68 hilo_tkinter.start()
69
70 while True:
71
72
73
74     with open('datos.txt', 'r') as archivo: #Los datos leídos desde la interfaz se guardan en
75         valor = str(archivo.read()) #Esto permite cambiar el color
76         if valor != '':
77             #separar los valores con comas (,) y convertirlos a flotantes
78             valor = valor.split(",")
79             Hmin,Smin,Vmin, Hmax,Smax,Vmax,sc = float(valor[0]),float(valor[1]),float(valor[2])
80             #Si esta vacío el Txt se deja como flotante = 0, sino genera error
81             valor = float(0)
```

```
main.py x
main.py > actualizar_imagen
1 import threading
2 import interfazGrafica
3 import cv2
4 import numpy as np
5 import rescaleFrame
6 import sys
7
8 cnt_Count = 0
9 def actualizar_imagen(HminIn,SminIn,VminIn,HmaxIn,SmaxIn,VmaxIn):
10     # Crea una máscara usando los valores de un canal en HSV
11     lower_color = np.array([HminIn, SminIn, VminIn])
12     upper_color = np.array([HmaxIn, SmaxIn, VmaxIn])
13     mask = cv2.inRange(hsv_image, lower_color, upper_color)
14
15     # Aplica la máscara a la imagen original
16     resultado = cv2.bitwise_and(original_image, original_image, mask=mask)
17     resultado = cv2.Canny(resultado, 50, 255)
18     # Aplica el filtro de mediana con un tamaño de kernel (ventana) específico
19     #resultado = cv2.medianBlur(resultado, 5) # El segundo argumento es el tamaño del kern
20     # Convierte la imagen HSV a RGB
21     im_rgb = cv2.cvtColor(resultado, cv2.COLOR_HSV2RGB)
22     # Define el kernel (elemento estructurante) para la erosión y la dilatación
23     kernelSz = np.ones((3, 3), np.uint8)
```





Modelo de negocio

- ▶ Ventas licencias
- ▶ Datos como servicios

Equipo



Christian Michael Florez
Ing. Automática industrial-
Programador



Carlos Andres Higon
Tecg. Telemática -Programador web



Gabriel Alejandro Fierro
Ing. Automática industrial-
Programador