



# Avance 1

# Sistema de Seguridad Perimetral

**Integrantes:** Ernesto García  
Daniel Ramírez

**Curso:** Proyecto IV

**Profesor:** Diego Aracena Pizarro



# Introducción

## Empresa

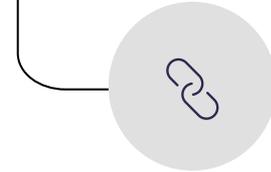
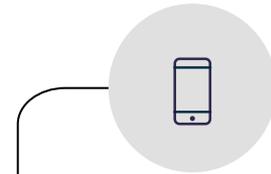
Que es  
WAKI Labs?

## Resumen del proyecto

Problema,  
solución,  
objetivos

## Diagramas

Subsistemas,  
Interacción,  
Comunicación



## Modelo de Datos

Y Diagrama E-R

## Implementación

Y pruebas del  
sistema

## Conclusión

Experiencias,  
aprendizajes



## Quienes son?

- Laboratorio de innovación fundado el 2016 en Arica
- Atiende a la industria de territorios extremos
- Espacio de coworking para emprendedores



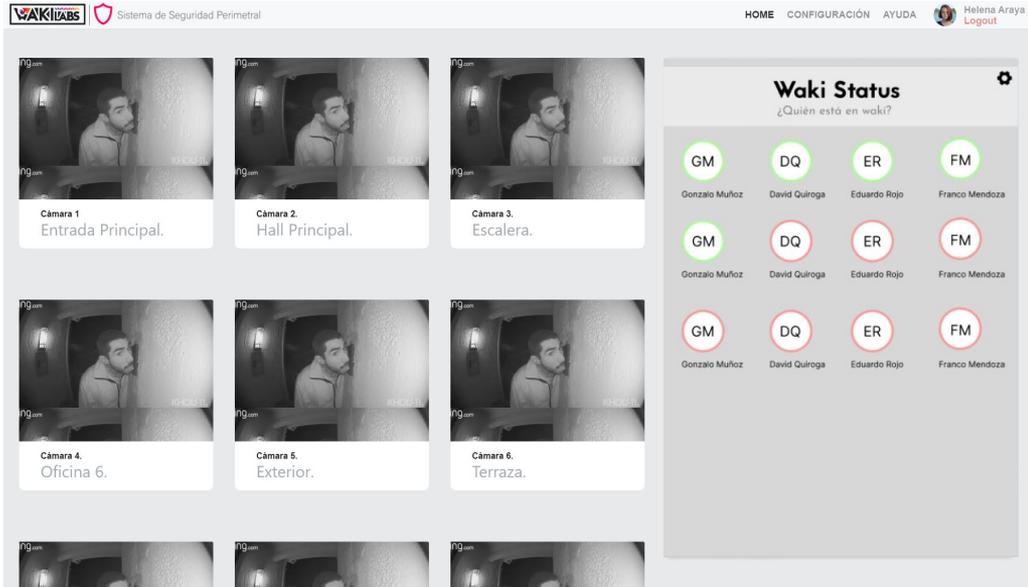
# Contexto del problema



Se necesita de un sistema nuevo que permita mantener los estándares de seguridad en sus instalaciones ya que:

- Sistema desactualizado
- Cámaras no funcionales
- Mala disposición de cámaras en el área
- Visibles sólo desde un televisor
- Posee una interfaz difícil de manejar

# Solución propuesta



Se planea crear un sistema que pueda replicar las funciones esenciales del sistema anterior.

- IGU accesible desde un servidor de página web
- Servidor de proceso de negocio en computador Linux en Waki Labs

Además, reposicionar aquellas cámaras que son desaprovechadas para conseguir una mejor cobertura del recinto y el área alrededor.

# Objetivo general



**Implementar** un sistema de seguridad y monitoreo remoto que pueda satisfacer las necesidades de seguridad de WAKI LABS.

# Objetivos específicos



1. Estudiar y definir la problemática.



2. Proponer la solución al problema y sus alternativas.



3. Desarrollar el sistema de acuerdo a la solución de software seleccionada.



4. Realizar las pruebas y análisis de resultados del sistema.

# Requisitos funcionales

ID	Definición
RF1	El sistema debe verificar al usuario.
RF2	El sistema debe permitir el ingreso de datos.
RF3	El sistema debe almacenar la información en un historial.
RF4	El sistema debe permitir ver todas las cámaras activas al mismo tiempo.
RF5	El sistema debe poder centrarse en una sola cámara.
RF6	El sistema debe poder alternar de cámara en cámara.

# Requisitos no funcionales

ID	Definición
RnF1	El sistema debe utilizar la paleta de colores de WAKILabs.
RnF2	El sistema debe ser programado en un entorno web.
RnF3	El sistema debe garantizar su funcionamiento en el navegador Mozilla Firefox.
RnF4	El sistema debe poder verse a través de cualquier navegador.
RnF5	El sistema debe tener una interfaz intuitiva y fácil de utilizar.
RnF6	El sistema deberá ser desarrollado utilizando el framework Angular.
RnF7	El sistema debe ser accesible para dispositivos móviles y sea "responsive".

# Organización del proyecto



Utilizado como un espacio de trabajo que permite subir archivos, tareas a realizar, bitácoras e informes.



Utilizado como un medio para dudas y consultas sobre el proyecto, organizar reuniones o informar a los otros miembros del equipo.



Utilizado como repositorio para la solución a desarrollar, a la cual sólo tiene acceso el equipo de trabajo.



# Herramientas del proyecto



Es una herramienta utilizada en el curso de proyecto para gestionar los proyectos de los estudiantes y realizar seguimiento de otros.



Es un editor de código fuente desarrollado por Microsoft. Cuenta con soporte para depuración de código y un sinnúmero de extensiones.



Es un Framework mantenido por Google, que sirve para desarrollar aplicaciones web de estilo SPA (Single Page Application).

# Herramientas del proyecto



Es un lenguaje de programación interpretado, multiparadigma. Es de tipo dinámico, multiplataforma y multipropósito.



Es una librería para Python que facilita el acceso a una base de datos relacional, así como las operaciones a realizar sobre la misma.

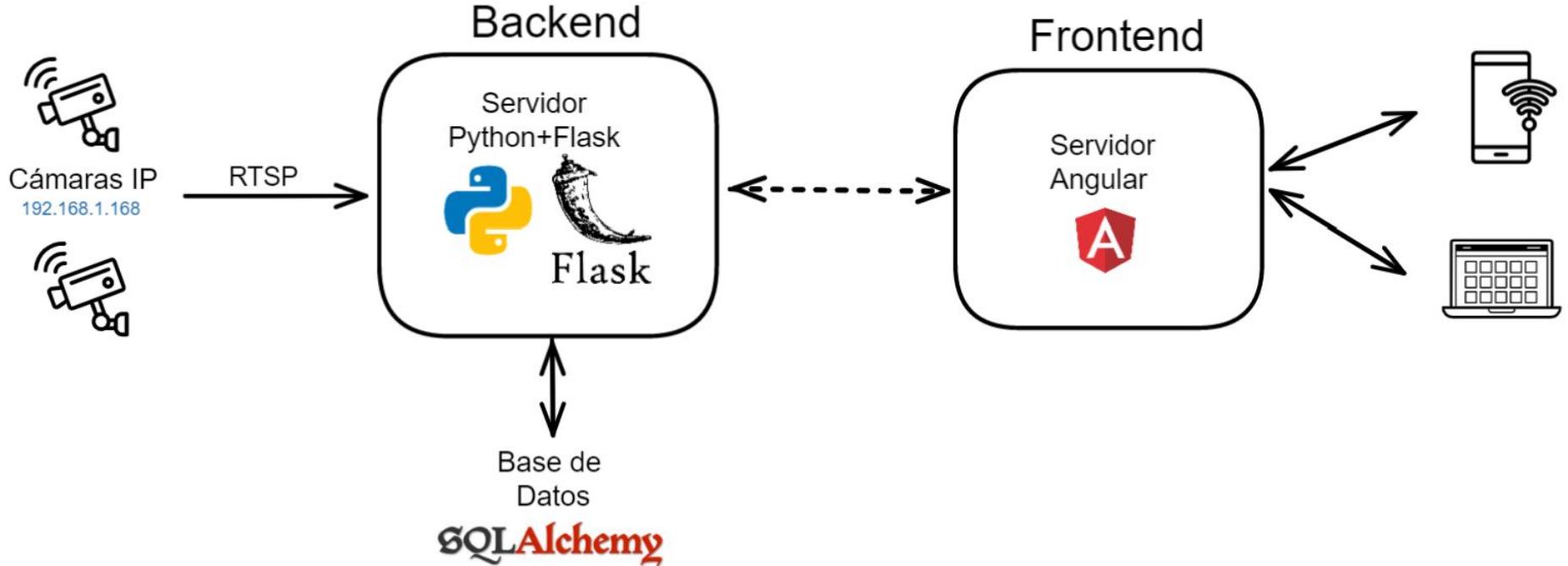
# Metodología

The image displays a collection of Scrum project management boards and a central graphic. The boards are arranged in a grid-like fashion, showing various stages of a project:

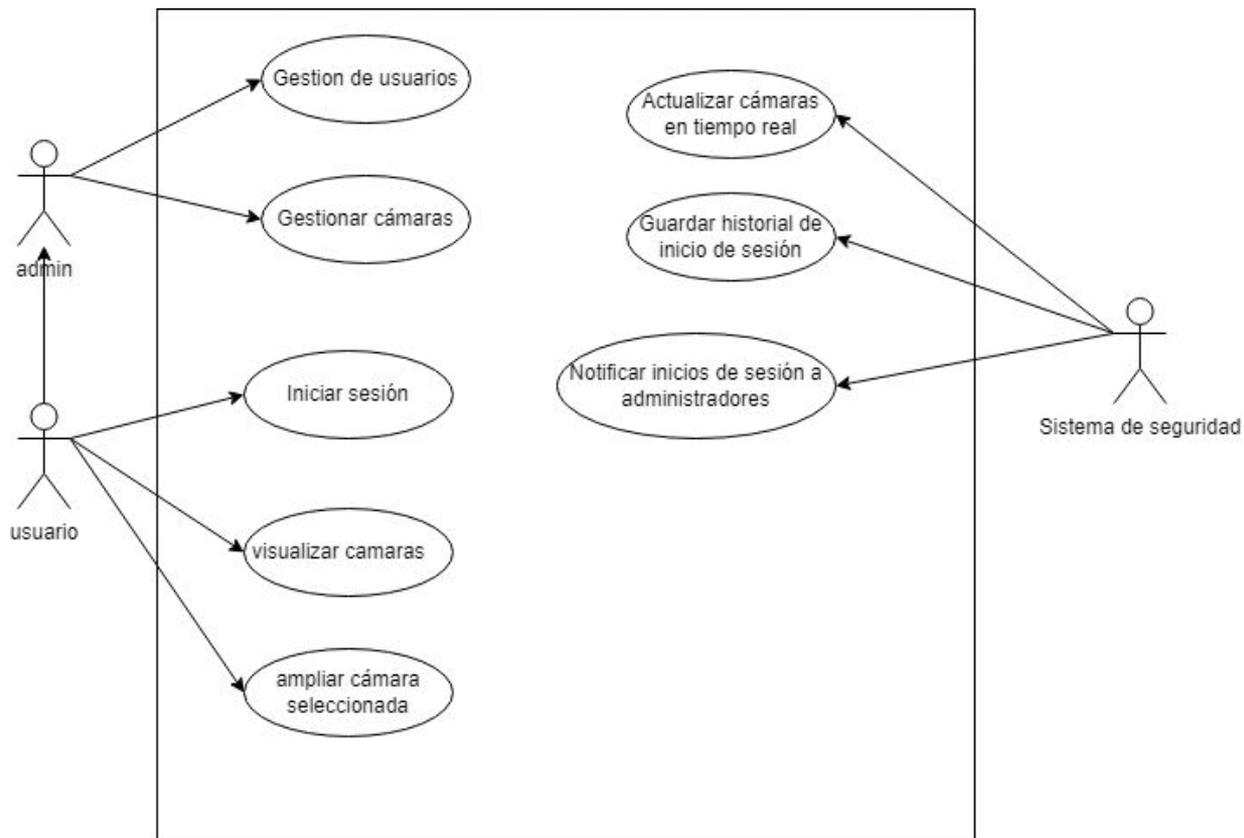
- Modelos de ejemplo:** A board with a pink header, showing a Gantt chart and a task card for 'Paleta de Colores (Solicitar a David)'.
- Backlog:** A board with a dark blue header, listing tasks like 'Realizar Diagrama ER y de Clases' and 'Investigar IoT'.
- Sprint Backlog - [Para Hacer]:** A board with a blue header, listing tasks for a sprint, including 'Reunión (5) 04-10-22' and 'Análisis de las necesidades y requerimientos del sistema'.
- Working On (En Proceso):** A board with a light blue header, showing a task card for 'Bitácoras de avance semanales'.
- Testing [Pruebas del Sistema]:** A board with a yellow header, showing a task card for 'Esperando feedback'.
- Bugs:** A board with a red header, showing a task card for 'Probar cámara VTECH'.
- Tareas Terminadas:** A board with a green header, showing a 'Done!' status and a list of completed tasks.

In the center, there is a large graphic with a teal background. On the left, a white rocket is launching. To the right of the rocket, the word **SCRUM** is written in large, bold, black capital letters. At the bottom right of the graphic, the text 'BEAGILE MYFRIEND' is visible.

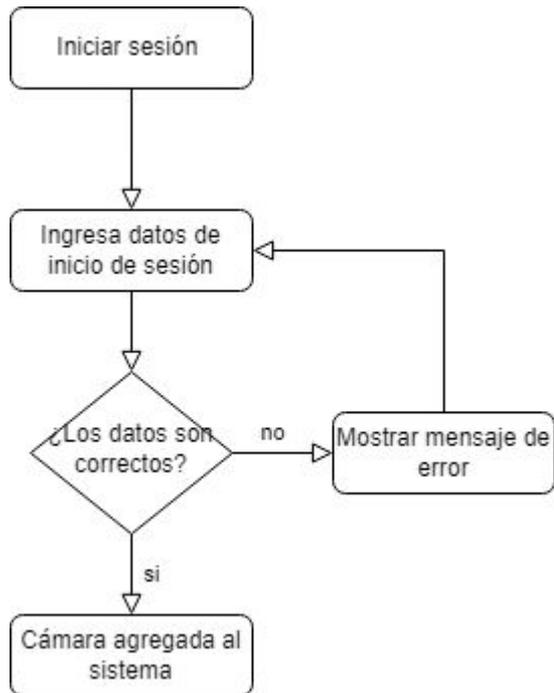
# Arquitectura del sistema



# Diagrama de casos de uso



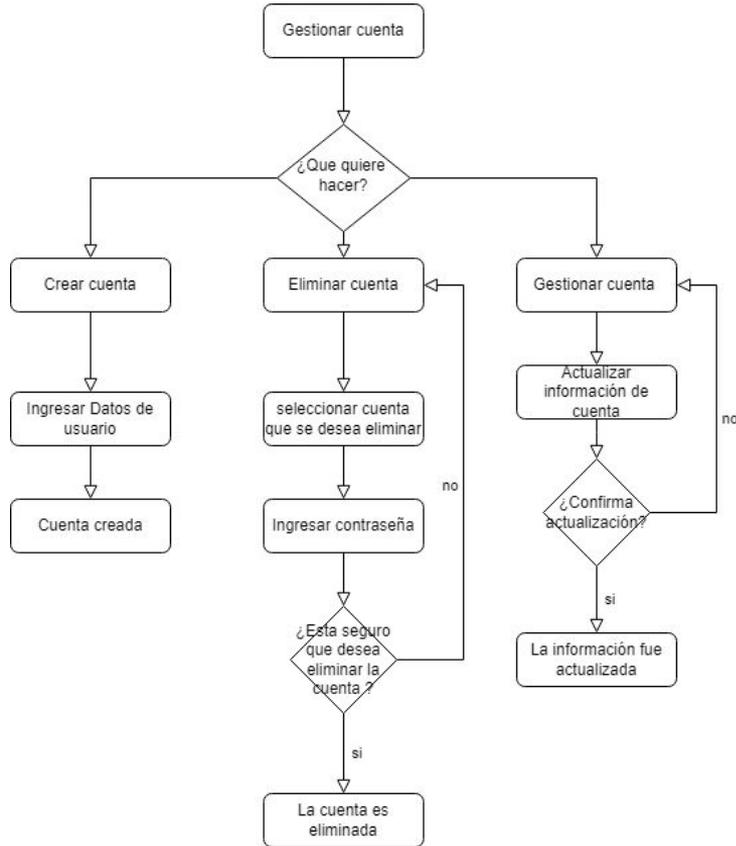
## Caso de uso: Inicio de sesión



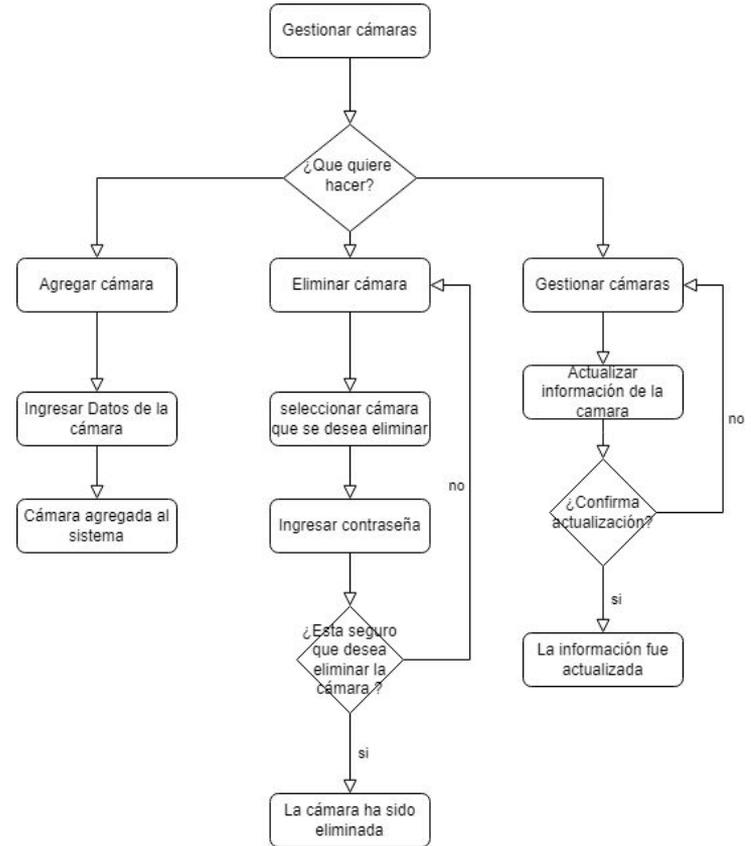
## Caso de uso: Visualizar cámaras



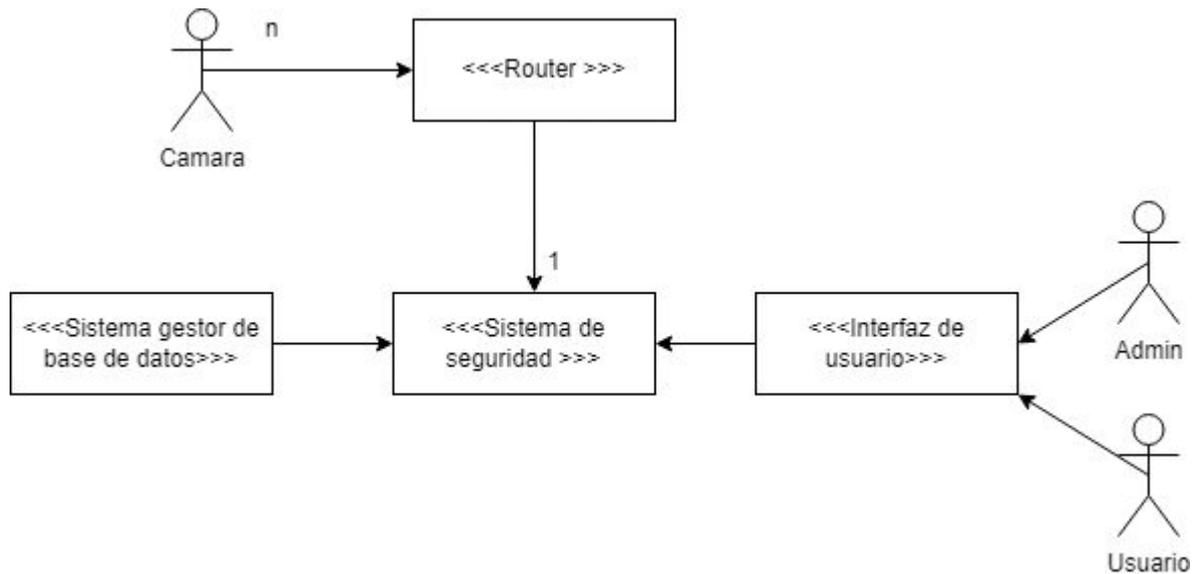
# Caso de uso: Gestionar cuentas



# Caso de uso: Gestionar cámaras



# Modelo de contexto



# Subsistemas



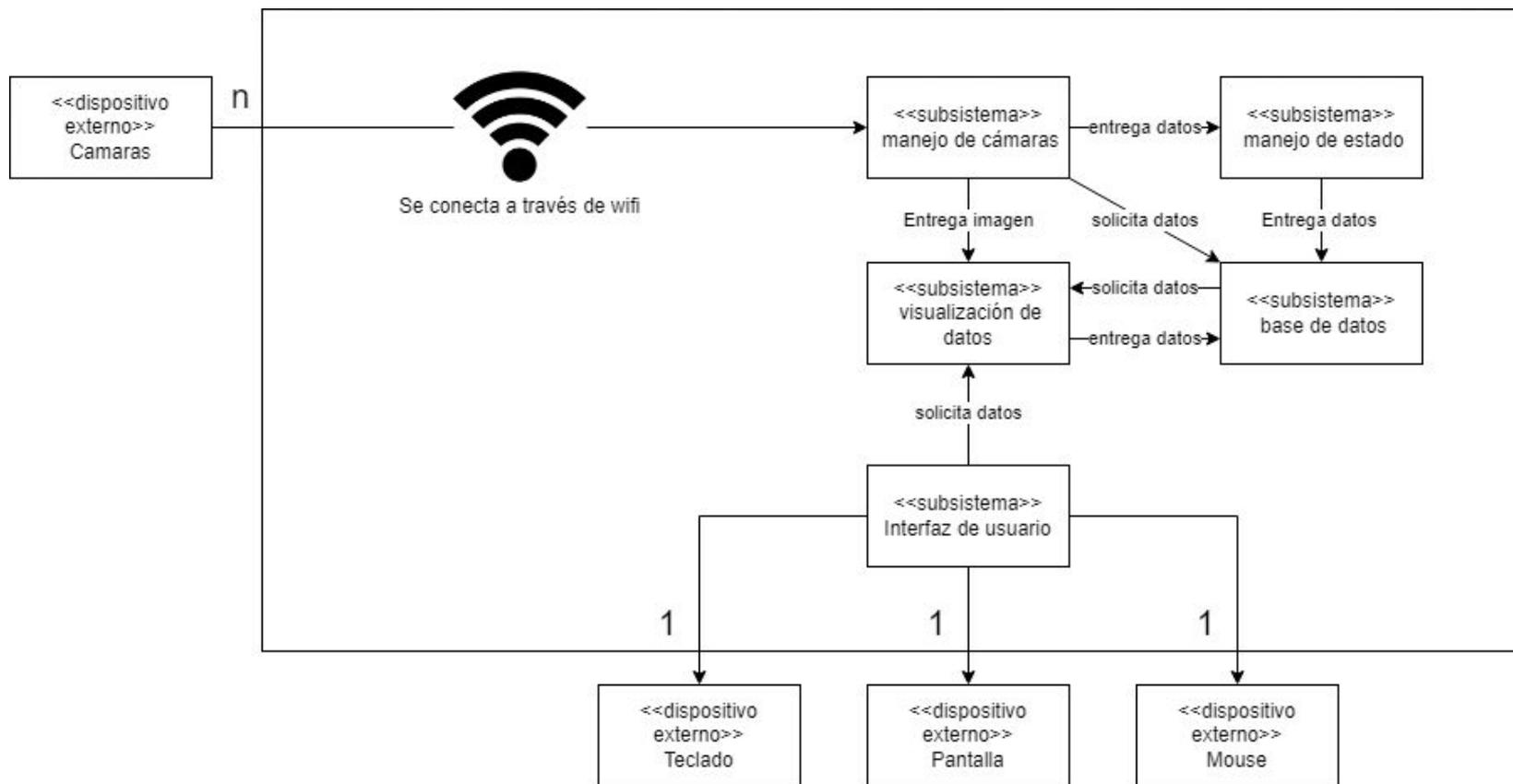
**Subsistema de manejo de manejo de cámaras:** Este es el encargado de gestionar las cámaras, conectarlas al sistema y permitir ver el contenido de estas.

**Subsistema de manejo de manejo de estado:** Este es el encargado de gestionar el estado de los trabajadores, en este caso es un booleano que simboliza si están o no dentro de Waki Labs.

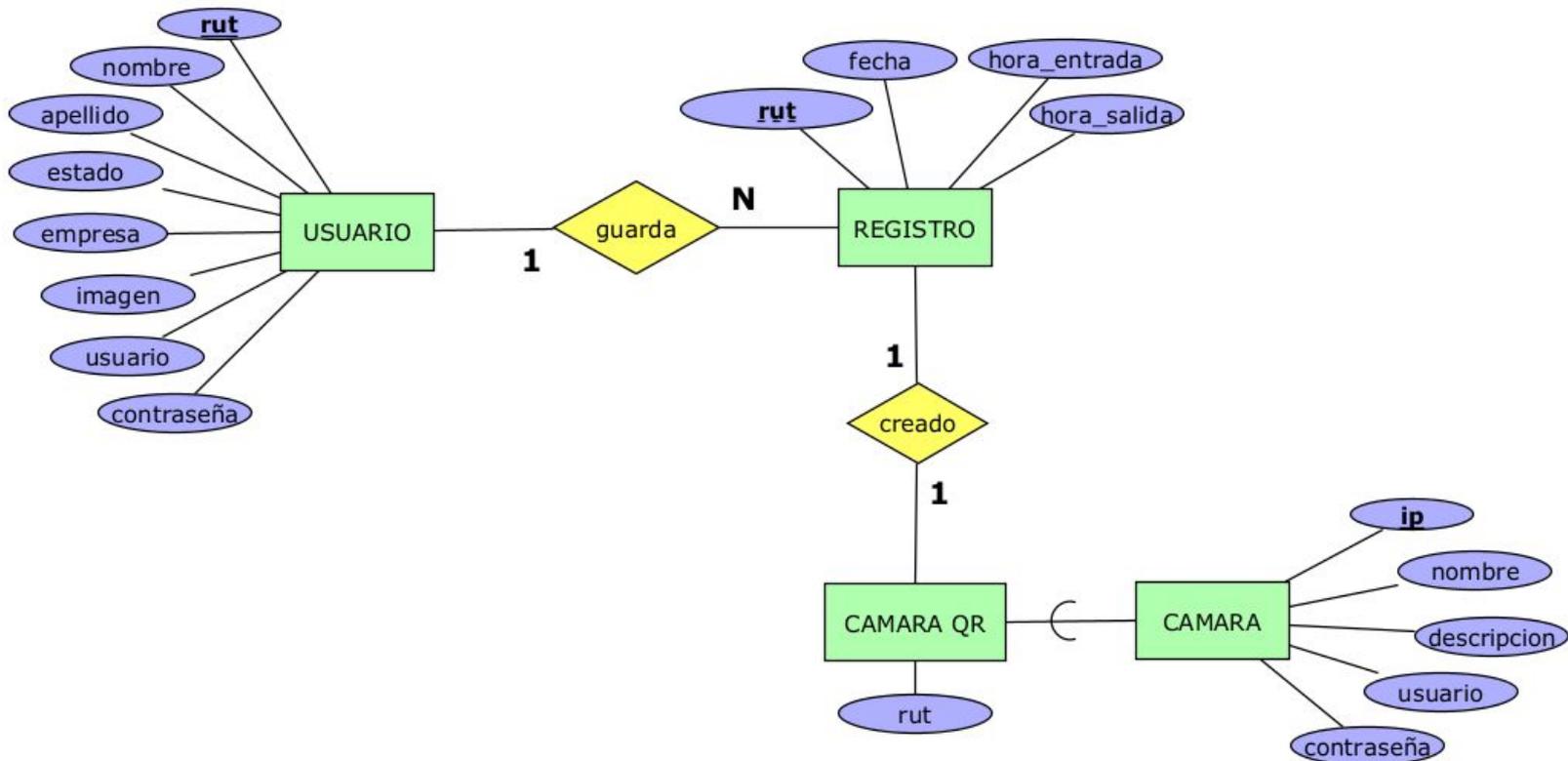
**Subsistema de base de datos:** Este es el encargado de gestionar los datos que utiliza el sistema, trabajadores, datos de las cámaras y estatus de los trabajadores.

**Subsistema de visualización de datos:** Este es el encargado de permitir al usuario ver en tiempo real los datos actualizados, las cámaras y los estatus de los trabajadores.

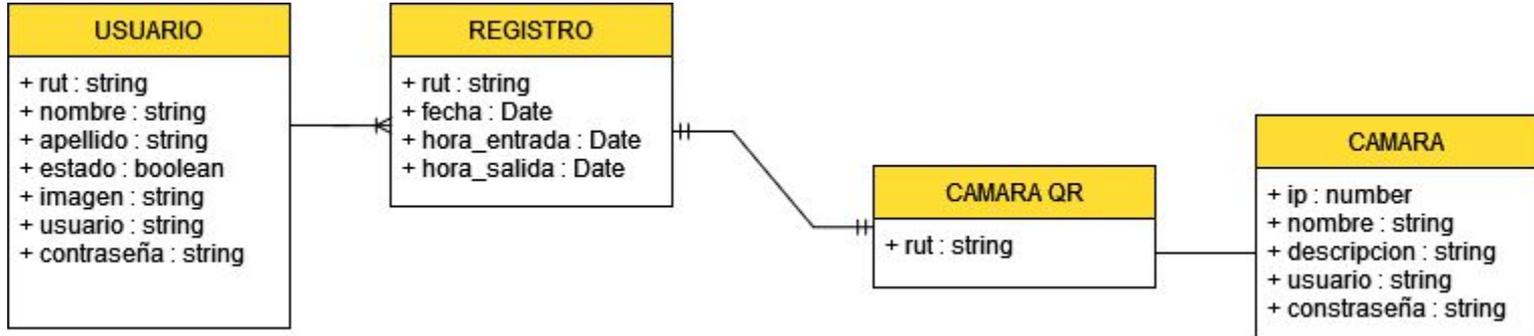
# Diagrama de Interacción



# Diagrama de Entidad-Relación



# Modelo de Datos



# Boceto para IGU

**WAKI LABS**  Sistema de Seguridad Perimetral

HOME CONFIGURACIÓN AYUDA  Helena Araya  
Logout

**Waki Status** 

¿Quién está en waki?

			
Gonzalo Muñoz	David Quiroga	Eduardo Rojo	Franco Mendoza
			
Gonzalo Muñoz	David Quiroga	Eduardo Rojo	Franco Mendoza
			
Gonzalo Muñoz	David Quiroga	Eduardo Rojo	Franco Mendoza

**Cámara 1**  
Entrada Principal.

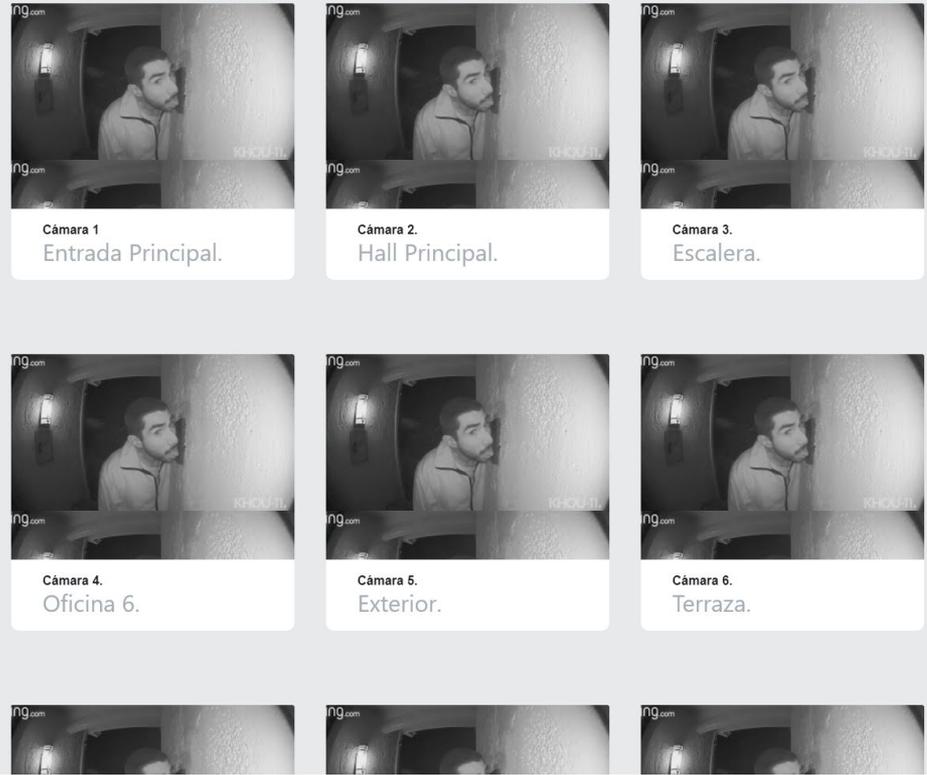
**Cámara 2.**  
Hall Principal.

**Cámara 3.**  
Escalera.

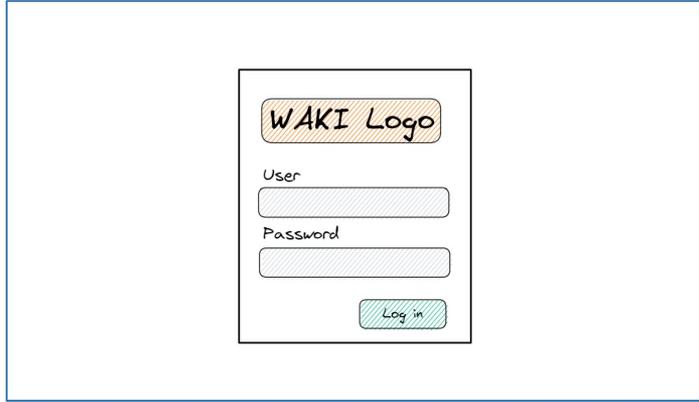
**Cámara 4.**  
Oficina 6.

**Cámara 5.**  
Exterior.

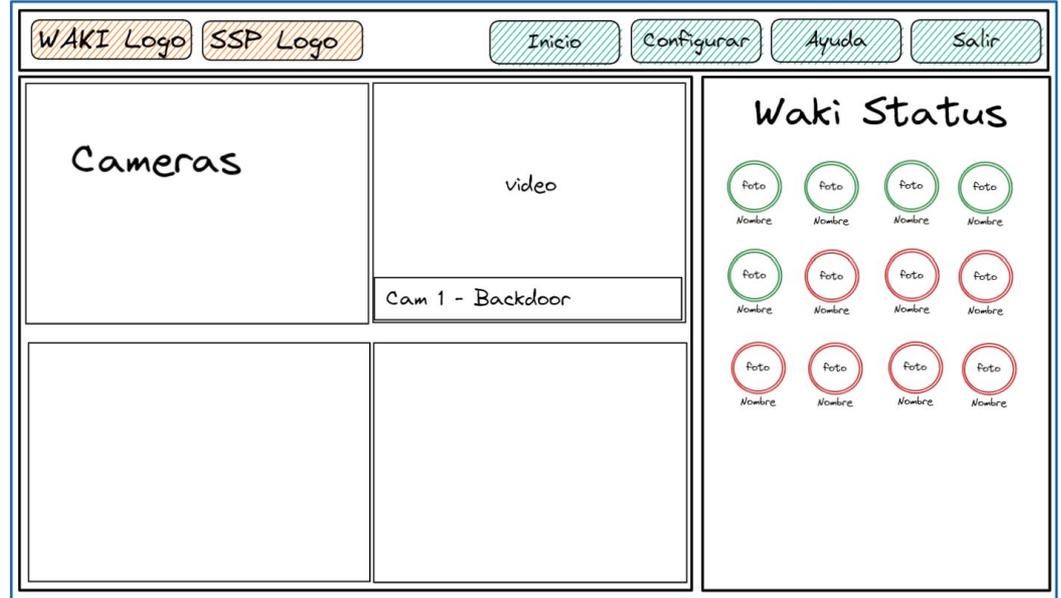
**Cámara 6.**  
Terraza.



# Bocetos para IGU



Vista Login



Vista Inicio

# Bocetos para IGU

WAKI Logo SSP Logo Inicio Configurar Ayuda Salir

Usuarios > Configurations ⚙️

Cámaras >

Nuevo usuario

Lista de usuarios / cámaras

Nombre / User

IP / Rut

Empresa

Cancelar / Editar Crear usuario

Vista Configuración

WAKI Logo SSP Logo Inicio Configurar Ayuda Salir

Ayuda

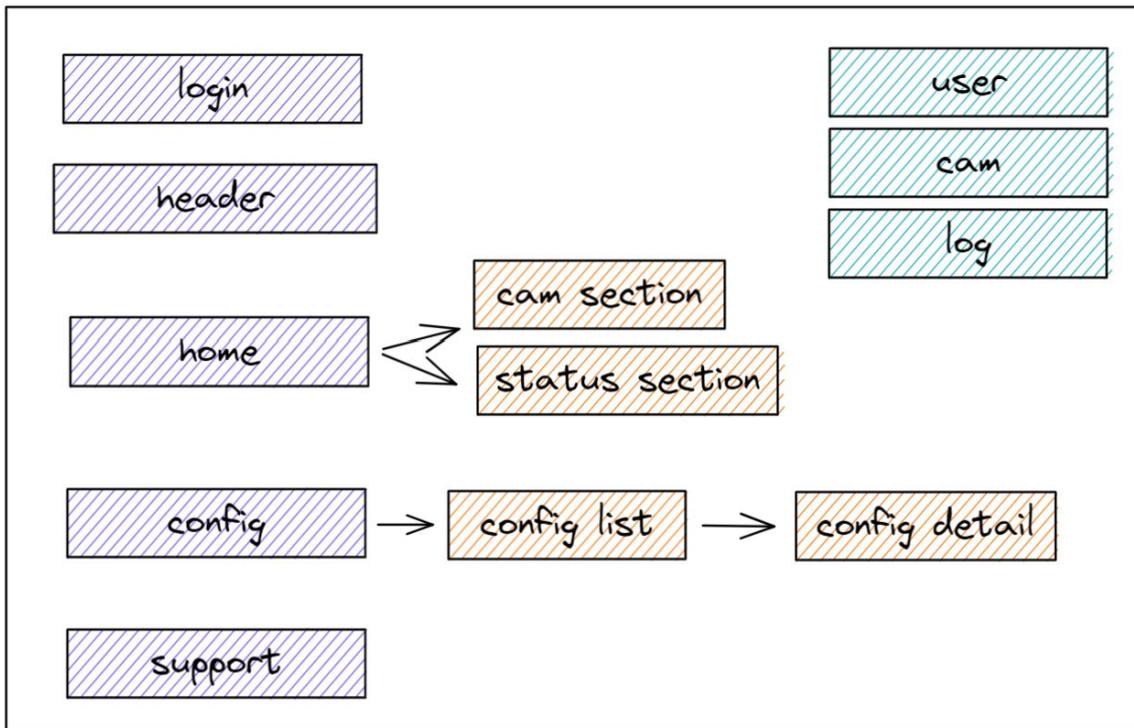
Guía para utilizar el sistema SSP

En esta sección explicaremos todas las funciones del sistema

1. Cámaras
- ...
2. Usuarios
- ...

Vista Ayuda

# Implementación del Sistema - Frontend



```
src
├── app
│   ├── config
│   ├── header
│   ├── home
│   ├── login
│   ├── support
│   ├── app.component.css
│   ├── app.component.html
│   ├── app.component.spec.ts
│   ├── app.component.ts
│   ├── app.module.ts
│   ├── camera.model.ts
│   └── user.model.ts
```

Componentes en Angular

Planeación de componentes

# Implementación del Sistema - Frontend

The screenshot displays the 'Waki Status' web application. The interface includes a top navigation bar with the 'WAKILABS' logo on the left and 'Inicio', 'Configuración', 'Ayuda', and a 'Salir' button on the right. The main content area is divided into two sections. On the left, under the heading 'Portón', there is a video feed titled 'Cámara del portón de Waki' showing a garage door with an 'EZVIZ' logo in the top-left corner and a timestamp '23-10-2022 15:20:03' in the bottom-left corner. On the right, under the heading 'Waki Status', there is a grid of six user status cards. Each card features a black silhouette icon, a name, and an affiliation. The cards are arranged in two columns and three rows. The first two rows have two cards each, and the third row has two cards. The colors of the cards are green for the first two rows and red for the third row.

Nombre	Afiliación
Ernesto Garcia	UTA
Eduardo Rojo	Wakilabs
David Quiroga	Wakilabs
Daniel Ramirez	UTA
Gonzalo Muñoz	Wakilabs
Patricio Arias	Wakilabs

# Implementación del Sistema - Frontend

The screenshot shows a web application interface for WAKILABS. At the top left is the WAKILABS logo. The top right navigation bar includes links for 'Inicio', 'Configuración', 'Ayuda', and a 'Salir' button. Below the navigation, there are two main sections: 'Configuración' on the left and 'Editar Cámara' on the right. The 'Configuración' section contains a green 'Nueva cámara' button and a 'Lista de cámaras' section with four entries: 'Portón' (Cámara del portón de Waki), 'Patio' (Cámara del patio de Waki), 'Entrada' (Cámara de la entrada de Waki), and 'Interior' (Cámara del interior de Waki). The 'Editar Cámara' section features a form with fields for 'Nombre', 'Descripción', and 'IP', and three buttons: 'Agregar' (green), 'Cancelar' (red), and 'Eliminar' (red).

WAKILABS

Inicio Configuración Ayuda Salir

Usuarios Cámaras

## Configuración

Nueva cámara

### Lista de cámaras

**Portón**  
Cámara del portón de Waki

**Patio**  
Cámara del patio de Waki

**Entrada**  
Cámara de la entrada de Waki

**Interior**  
Cámara del interior de Waki

**Esquina**  
Cámara de la esquina de Waki

## Editar Cámara

Nombre

Descripción

IP

Agregar Cancelar Eliminar

# Implementación del Sistema - Backend

```
models.py > UserModel > json
1 from flask_sqlalchemy import SQLAlchemy
2 db = SQLAlchemy()
3
4 class UserModel(db.Model):
5     __tablename__ = 'users'
6     rut = db.Column(db.String(10), primary_key=True)
7     name = db.Column(db.String(20))
8     last_name = db.Column(db.String(20))
9     business = db.Column(db.String(50))
10    image = db.Column(db.String(255))
11    status = db.Column(db.Integer())
12    user = db.Column(db.String(20))
13    pwd = db.Column(db.String(20))
14
15    def __init__(self, rut, name, last_name, business, image, status, user, pwd):
16        self.rut = rut
17        self.name = name
18        self.last_name = last_name
19        self.business = business
20        self.image = image
21        self.status = status
22        self.user = user
23        self.pwd = pwd
24
25    def json(self):
26        return {"rut":self.rut,
27                "name":self.name,
28                "last_name":self.last_name,
29                "business":self.business,
30                "imagePath":self.image,
31                "status":self.status,
32                "user":self.user,
33                "pwd":self.pwd
34            }
```

Creando base de datos

```
camStreaming.py > ...
1 import cv2
2 import os
3 import numpy as np
4 import pyzbar.pyzbar as pyzbar
5
6 # Parametros de la cámara
7 user = 'admin'
8 pwd = ' '
9 ip = '192.168.1.89'
10 port = '554'
11
12 os.environ["OPENCV_FFMPEG_CAPTURE_OPTIONS"] = "rtsp_transport;udp"
13
14 url = 'rtsp://{user}:{pwd}@{ip}:{port}/h.264'.format(user,pwd,ip,port)
15 print('Conectandose a: ' + url)
16
17 cap = cv2.VideoCapture(url, cv2.CAP_FFMPEG)
18
19 font = cv2.FONT_HERSHEY_PLAIN
20 while True:
21     _, frame = cap.read()
22
23     decodedObjects = pyzbar.decode(frame)
24     for obj in decodedObjects:
25         #print("Data", obj.data)
26         cv2.putText(frame, str(obj.data), (50, 50), font, 2,
27                     (255, 0, 0), 3)
28
29         cv2.imshow('VIDEO', frame)
30
31         if cv2.waitKey(1) & 0xFF == ord('q'):
32             break
33
34 cap.release()
35 cv2.destroyAllWindows()
```

Leyendo QR con OpenCV

# Pruebas del Sistema



--Language-- ▾

**live** setup

**console**

 record

 play



 zoom 

stream switch

Audio : On

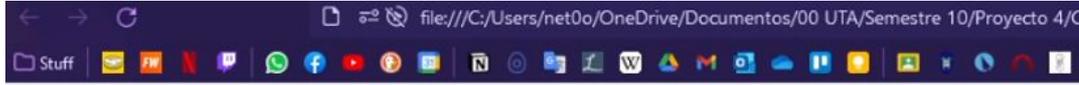
Snapshot

main stream



Adobe Flash Player is no longer supported

# Pruebas del Sistema



# Conclusiones



Extensa investigación de los temas no conocidos para comenzar a programar



Mantener información con el cliente y tener buena disponibilidad



Problemas de desarrollo, búsqueda de soluciones



Definir un límite de requerimientos y respetarlo

**Gracias  
por su  
atención**

