



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

## Sistema de prueba de vida para login biométrico usando modelos de machine learning

Cáceres Erraez Cristóbal Alejandro

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Sistemas e Informática

Trabajo de titulación, previo a la obtención del título de Ingeniero en Sistemas e Informática

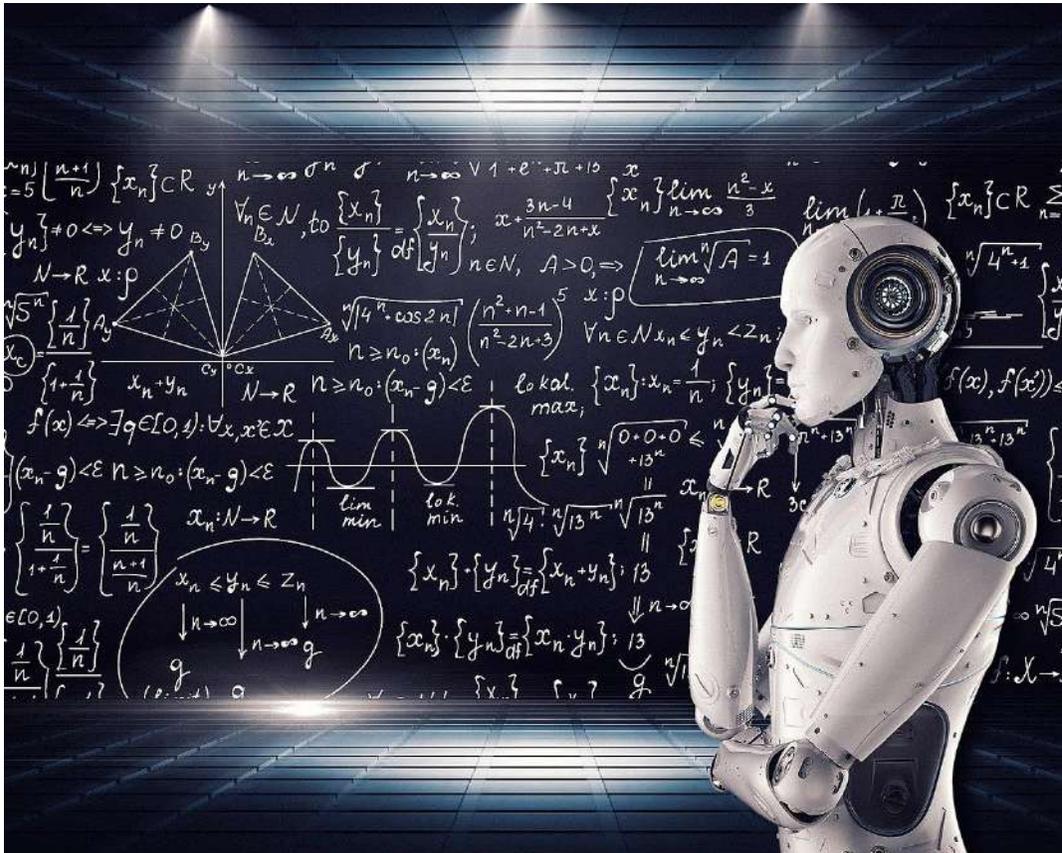
Msc. Delgado Rodríguez, Ramiro Nanac

24 de agosto del 2022

# AGENDA

- 1 • Introducción, Objetivos y Alcance
- 2 • Descripción del proyecto
- 3 • Desarrollo del Sistema
- 4 • Análisis de resultados
- 5 • Conclusiones y Recomendaciones

# INTRODUCCIÓN



Determinar si una persona esta viva o no, para un humano puede ser relativamente fácil... pero para un programa?

Es aquí donde el uso de modelos de inteligencia artificial nos facilitan el desarrollo de diversas soluciones, en este caso una prueba de vida.

Se va a tocar superficialmente el gran campo de la Inteligencia Artificial para poner en contexto la potencia de las herramientas que se encuentran en nuestras manos

# OBJETIVOS

## **General:**

Diseñar un sistema de prueba de vida para login biométrico en web usando modelos de machine learning.

## **Específicos:**

- Investigar modelos de machine learning que permitan el reconocimiento facial.
- Investigar sobre el uso del paradigma de servicios web para la codificación del sistema web
- Diseñar una solución que permita la integración de los servicios de prueba de vida para el proveedor de identidades y accesos keycloak.
- Realizar pruebas y evaluar resultados.

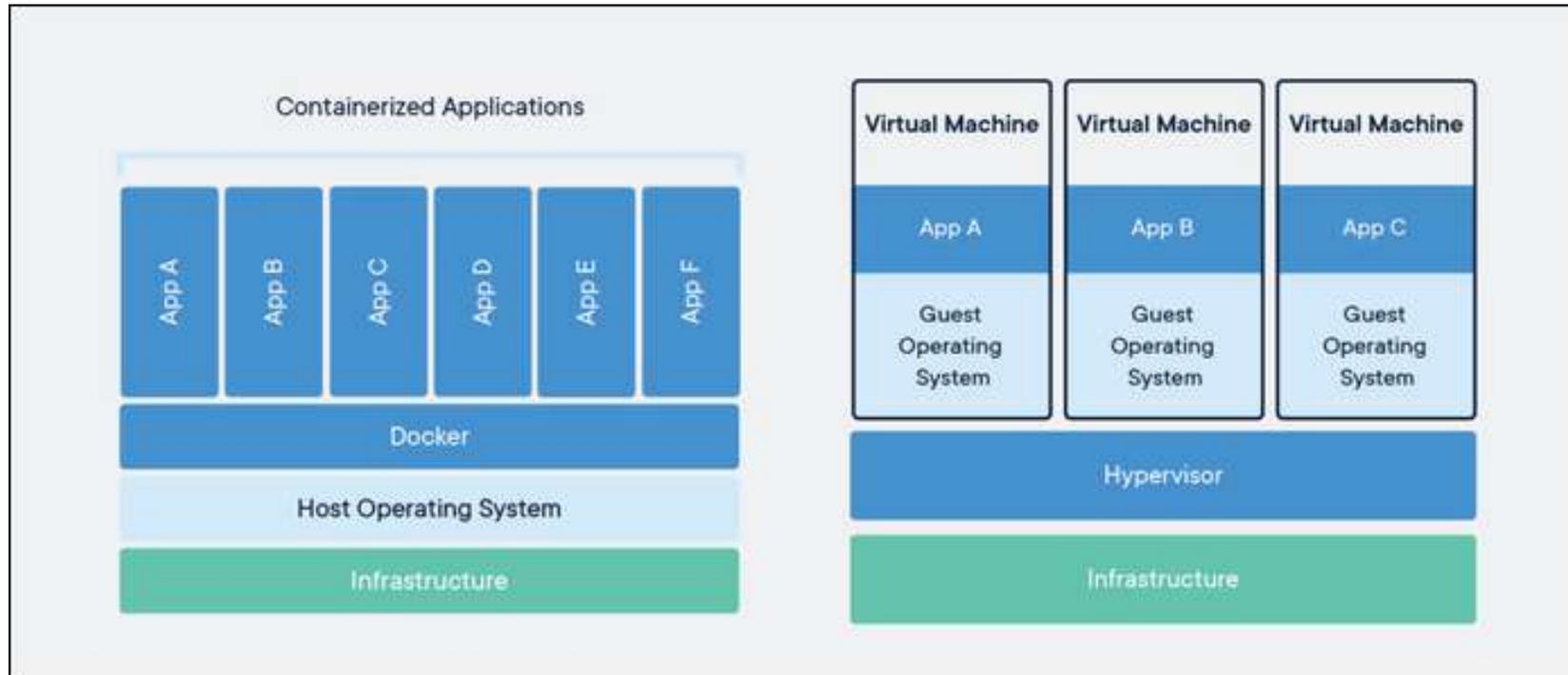
# ALCANCE



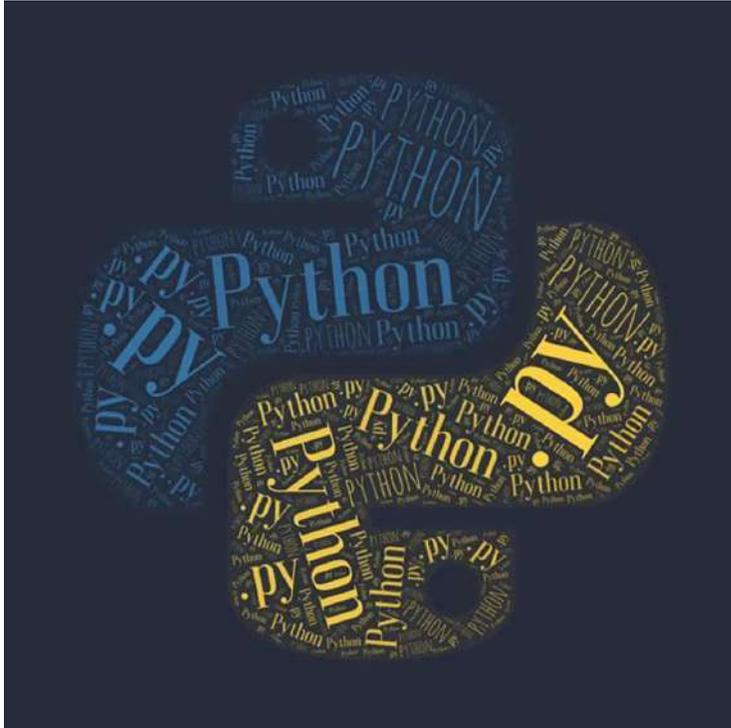
Se planteó como alcance original el desarrollo de una API que contenga la lógica del sistema de prueba de vida para login biométrico y una página web para poder realizar las pruebas del funcionamiento correcto de la misma.

Adicional al alcance original se generó la integración en el proveedor de identidades keycloak para generar un flujo funcional de webs securizadas.

# DOCKER



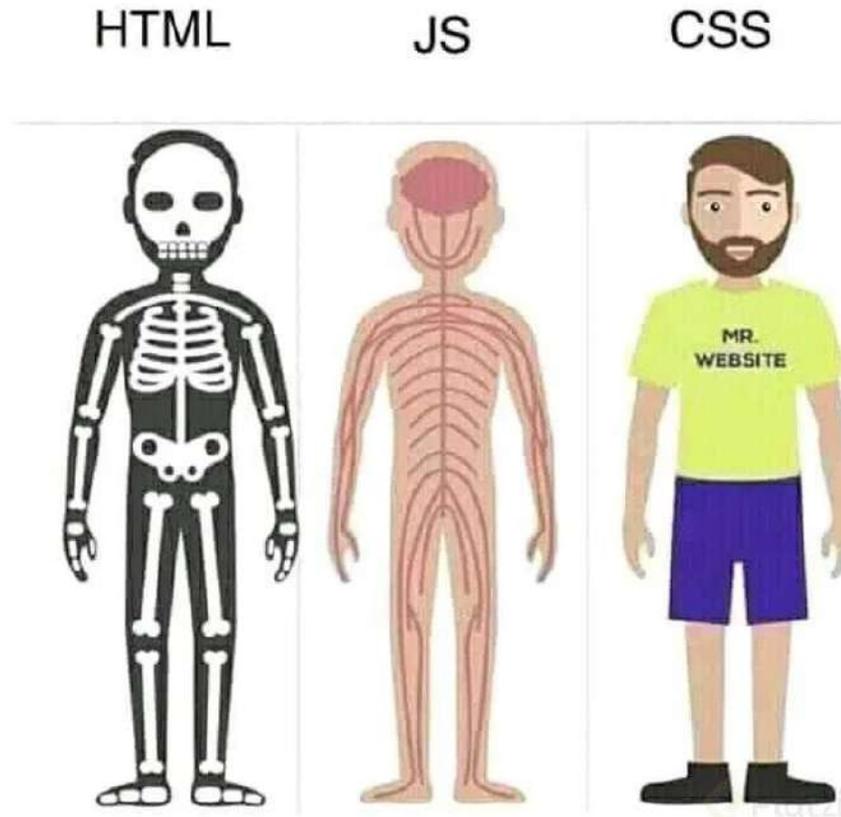
# PYTHON



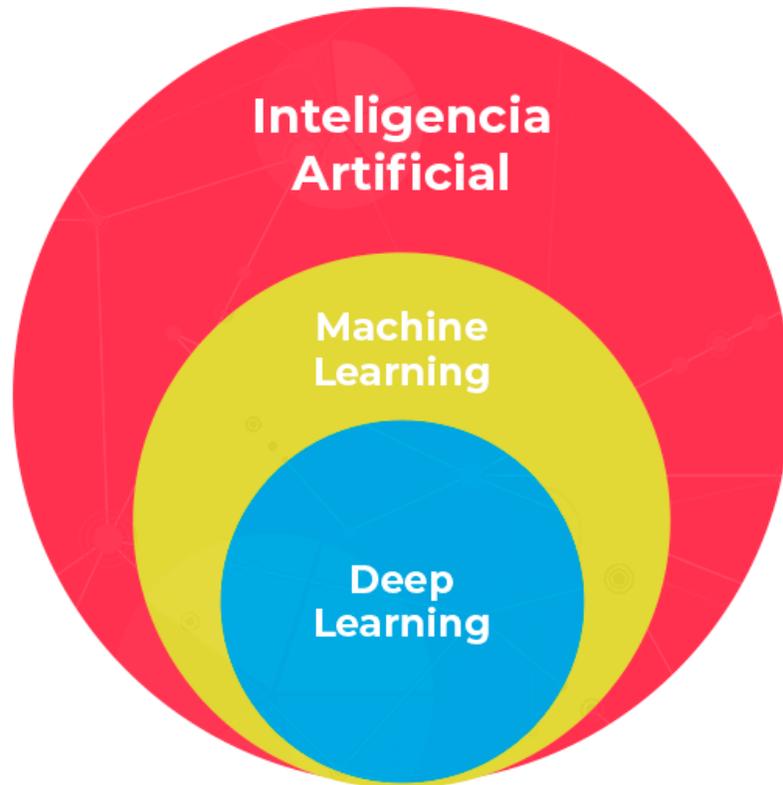
“Python es un lenguaje de programación que permite trabajar rápidamente e integrar sistemas de manera más efectiva.” (Python, 2021)

En el campo de la Inteligencia Artificial es muy común el uso de este lenguaje para el desarrollo de soluciones, por la facilidad y gran cantidad de información que se puede encontrar en internet es el lenguaje que se utilizó en el desarrollo del API de este proyecto

# HTML, CSS, JS

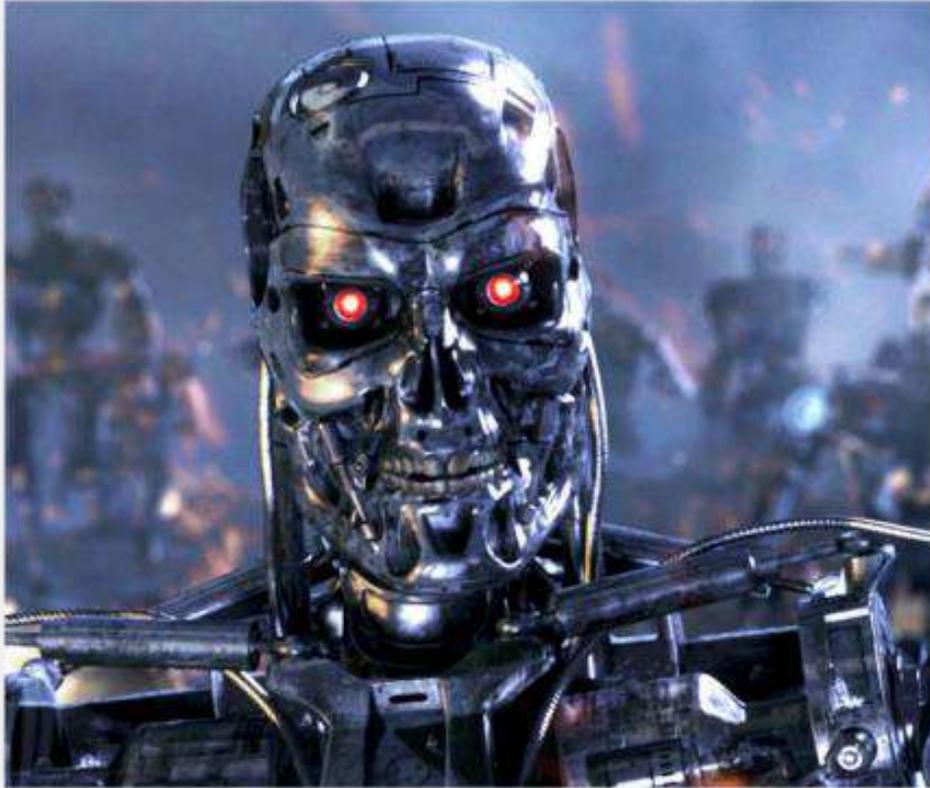


# INTELIGENCIA ARTIFICIAL



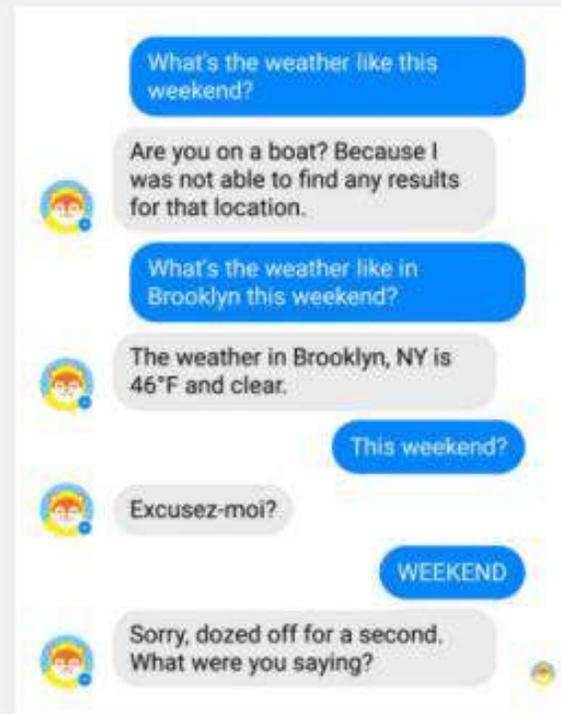
- **IA:** Combinación de algoritmos planteados con el propósito de crear máquinas que presenten las mismas capacidades que el ser humano.
- **Machine Learning:** Rama de la Inteligencia artificial (IA) que estudia como dotar a las máquinas de capacidad de aprendizaje
- **Deep Learning:** algoritmo automático jerárquico que emula el aprendizaje humano con el fin de obtener ciertos conocimientos.

# TIPOS DE INTELIGENCIA ARTIFICIAL



strong AI

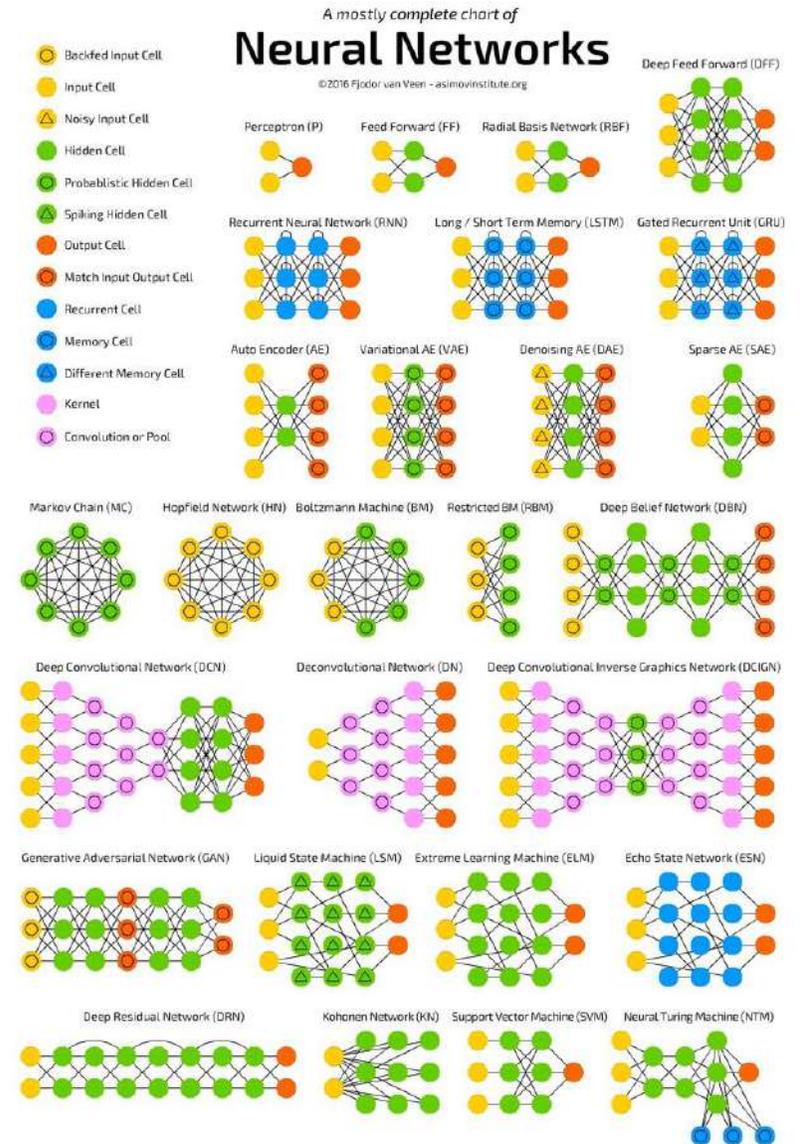
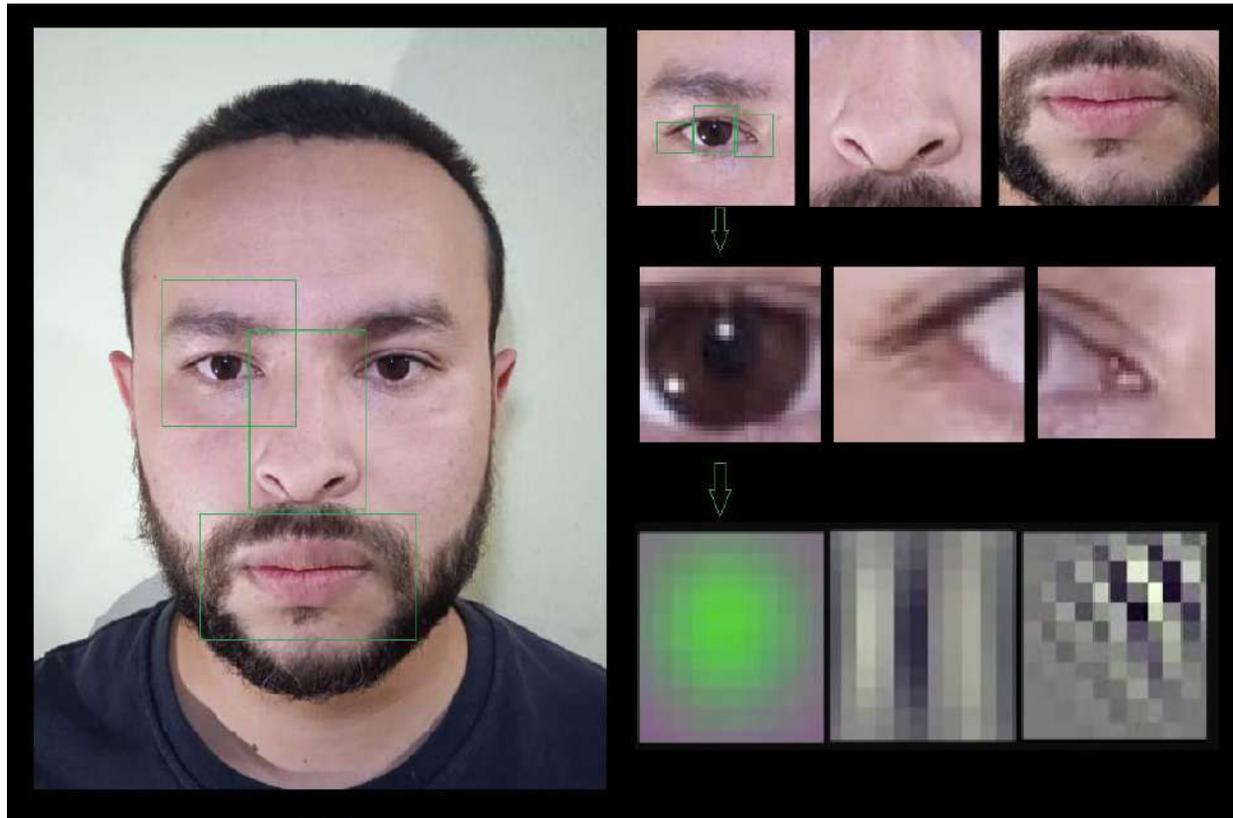
vs



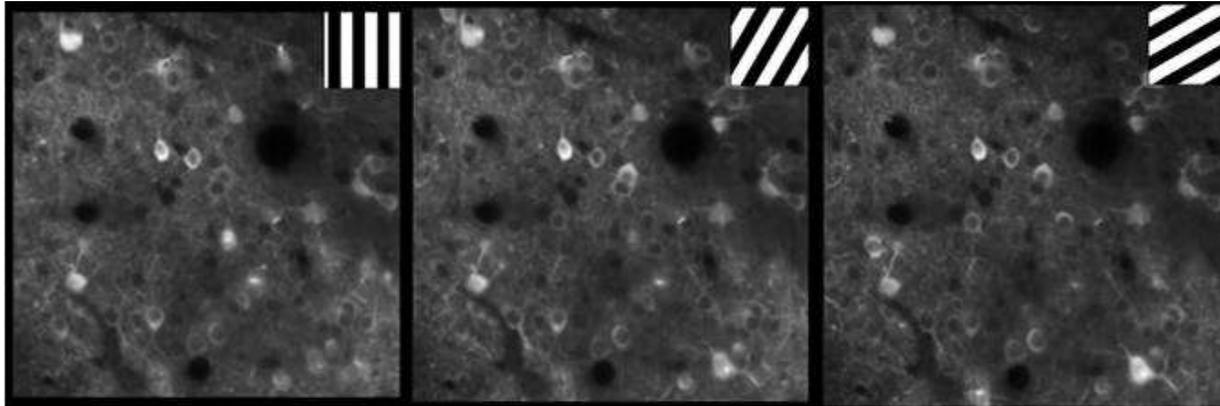
<https://chatbot.fail/>

weak AI

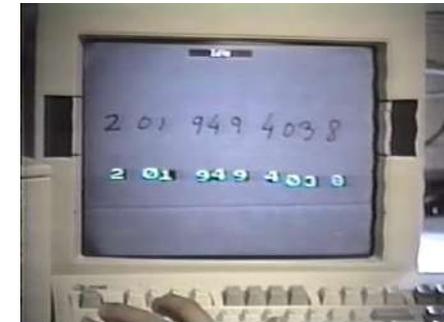
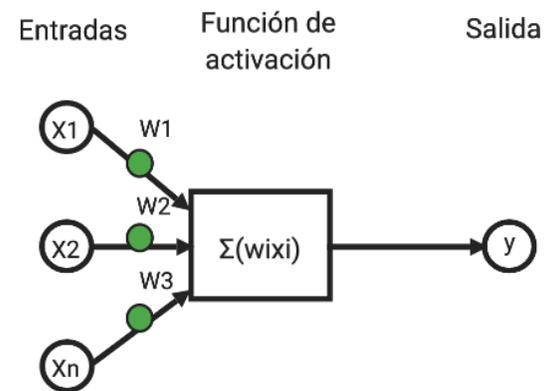
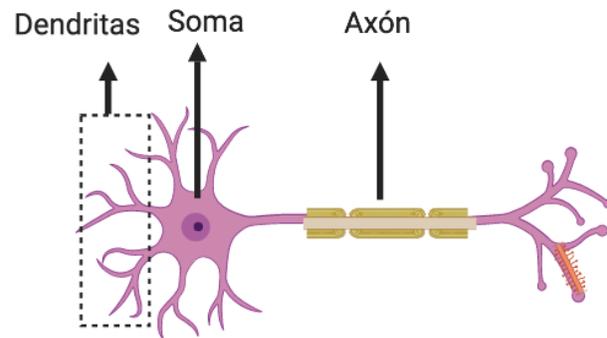
# MACHINE LEARNING



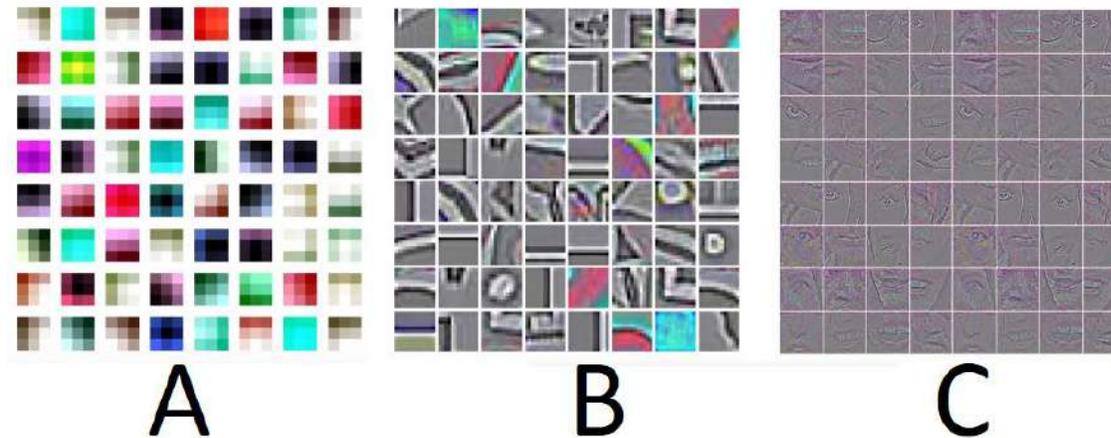
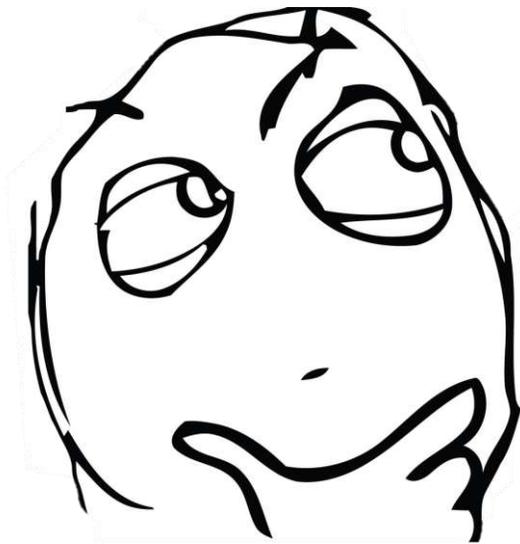
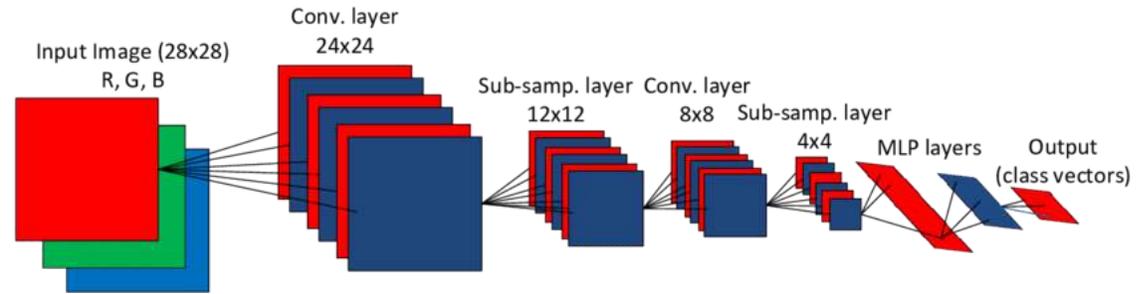
# REDES CONVOLUCIONALES



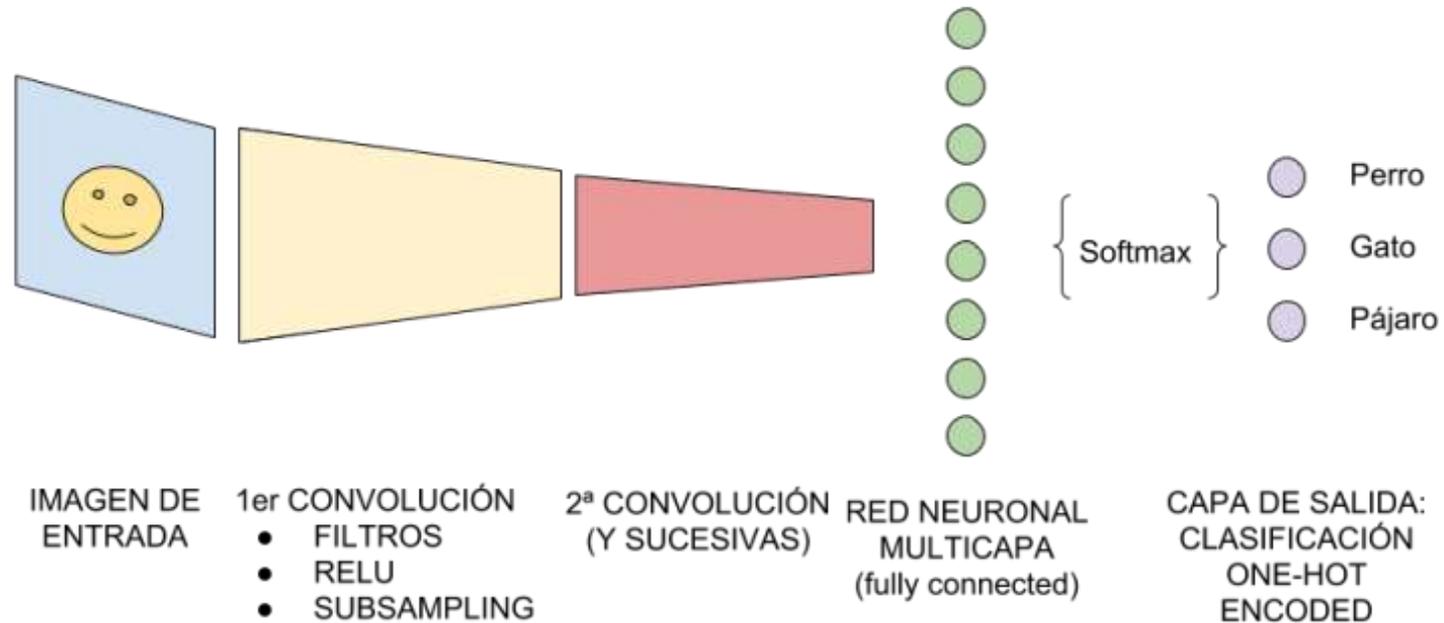
Yann Le Cun



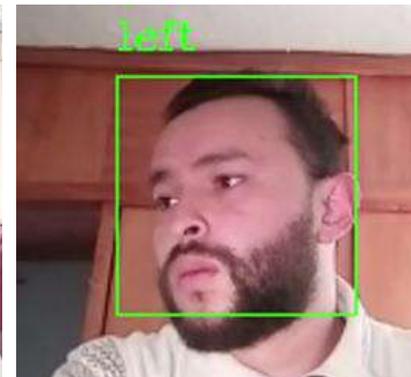
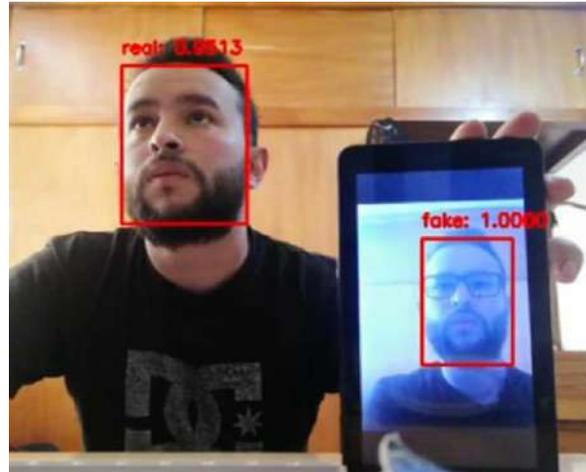
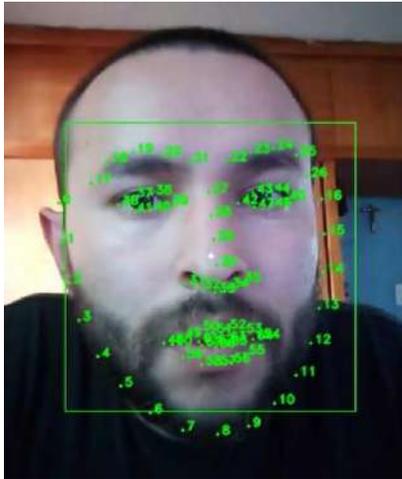
# REDES CONVOLUCIONALES



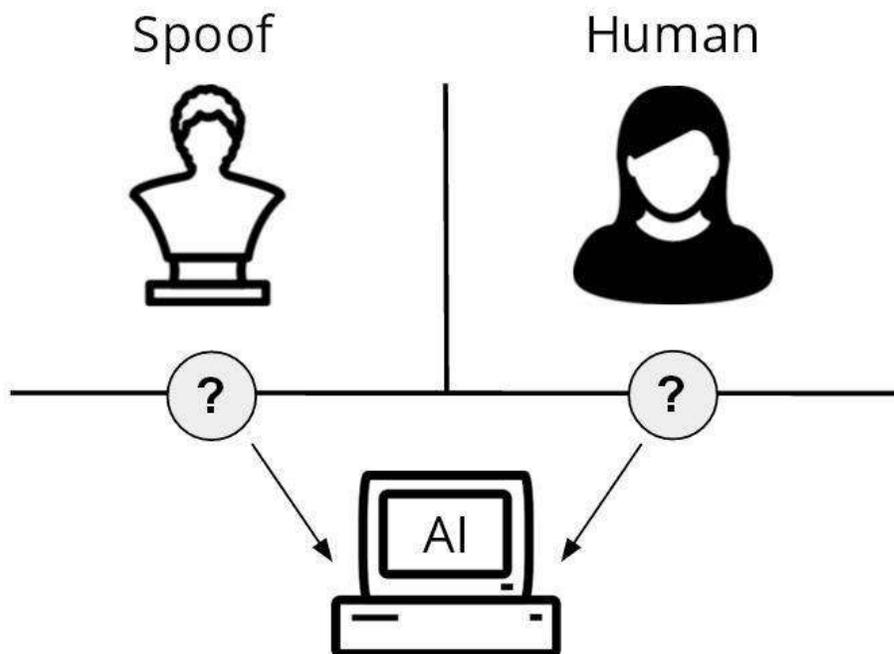
# MODELOS DE INTELIGENCIA ARTIFICIAL



# MODELOS DE INTELIGENCIA ARTIFICIAL



# PRUEBA DE VIDA

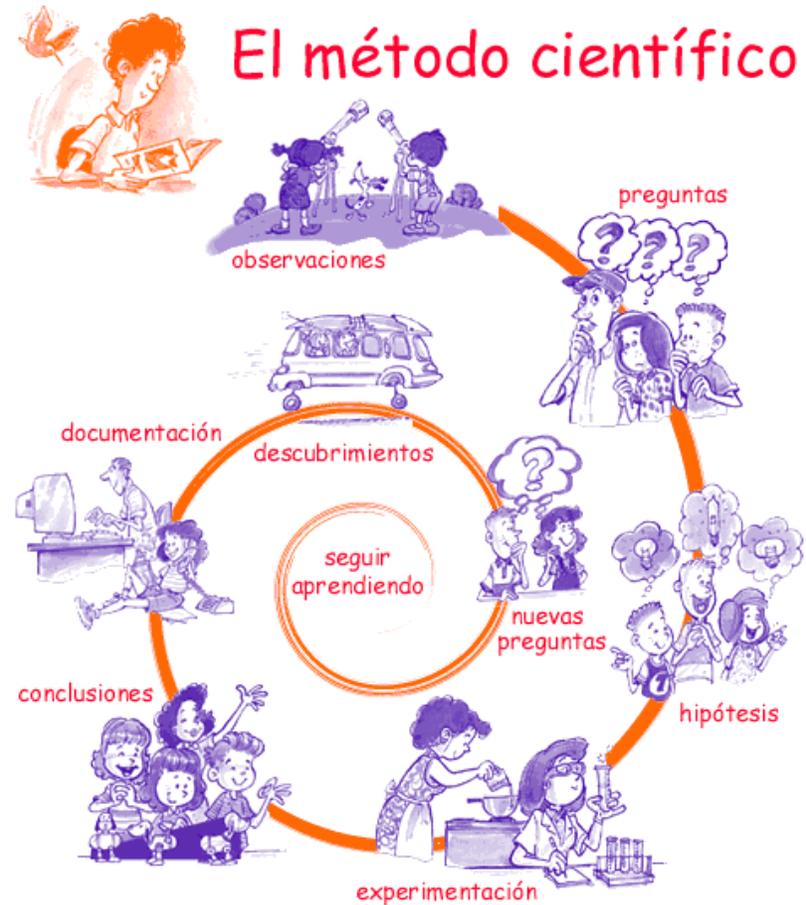


Tipo de artefacto	Descripción	Ejemplo
Nivel 1 (A)	Fotos digitales y en papel de alta resolución, videos de desafío / respuesta de alta definición y máscaras de papel.	
Nivel 2 (B)	Muñecos realistas disponibles comercialmente y máscaras 3D de resina, látex y silicona usadas por humanos con un precio inferior a \$ 300.	
Nivel 3 (C)	Máscaras 3D ultrarrealistas hechas a medida, cabezas de cera, etc., con un costo de creación de hasta \$ 3,000.	

Tipo de derivación	Descripción	Ejemplo
Nivel 4	Descifre y edite el contenido de un 3D facemap™ para que contenga datos sintéticos no recopilados de la sesión, haga que el servidor procese y responda con Liveness Success.	
Nivel 5	Toma el control de la alimentación de la cámara e inyecta fotogramas de video capturados previamente o una marioneta falsa que da como resultado que <u>facetec AI</u> responda con "Liveness Success".	

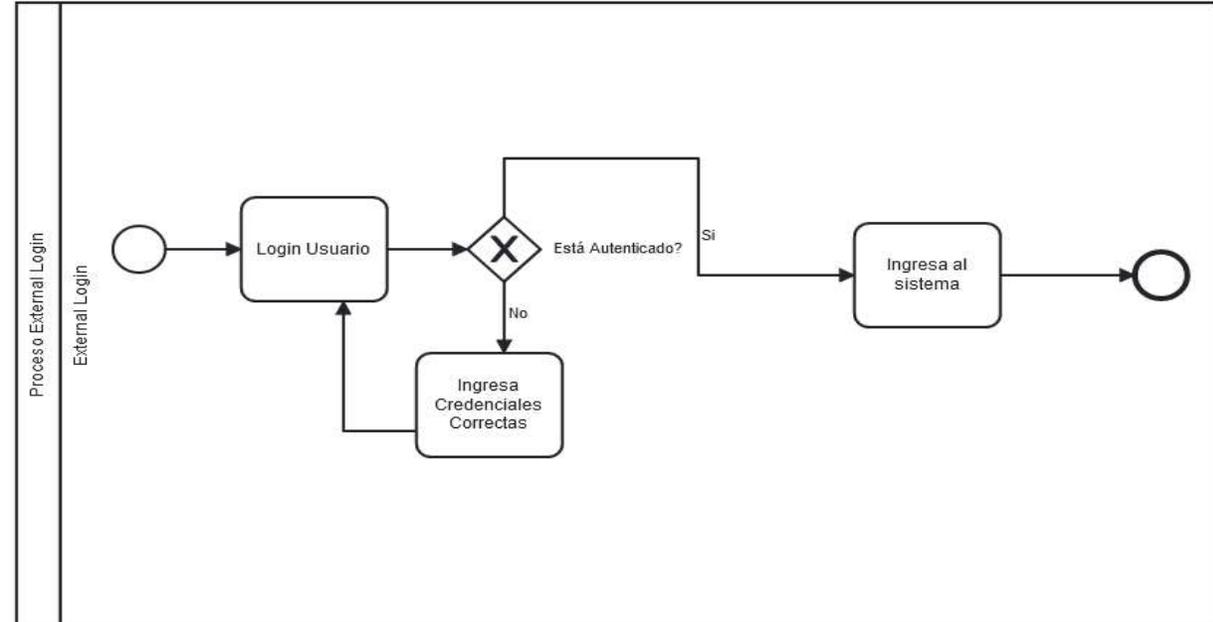
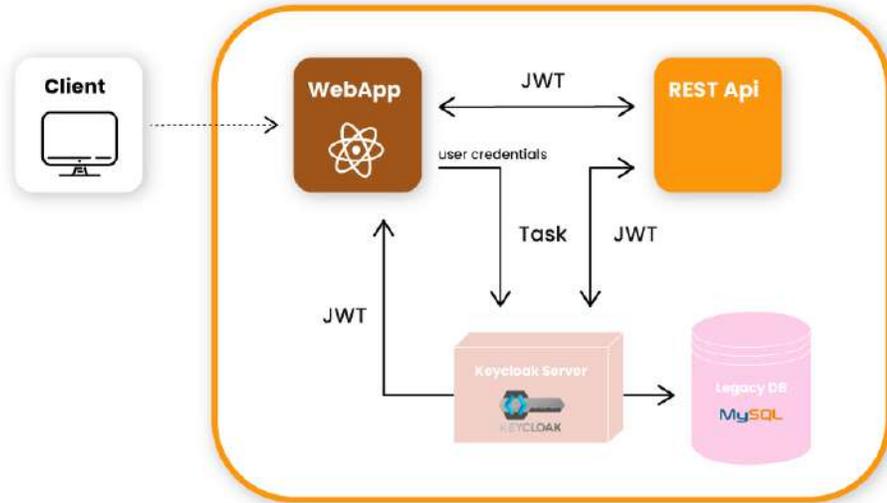
Descripción del proyecto

# METODOLOGÍA

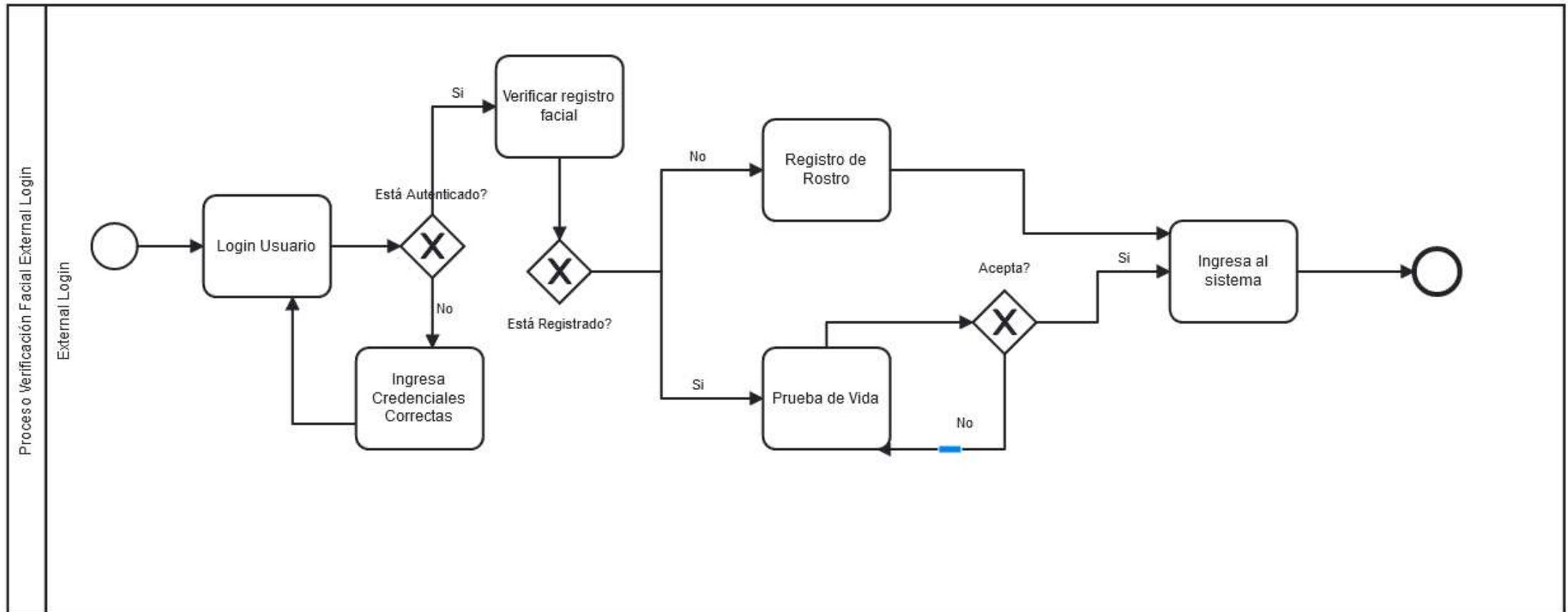


Desarrollo del sistema

# PUNTO DE PARTIDA



# PUNTO OBJETIVO



# DESARROLLO DEL SISTEMA

Subtareas ... +

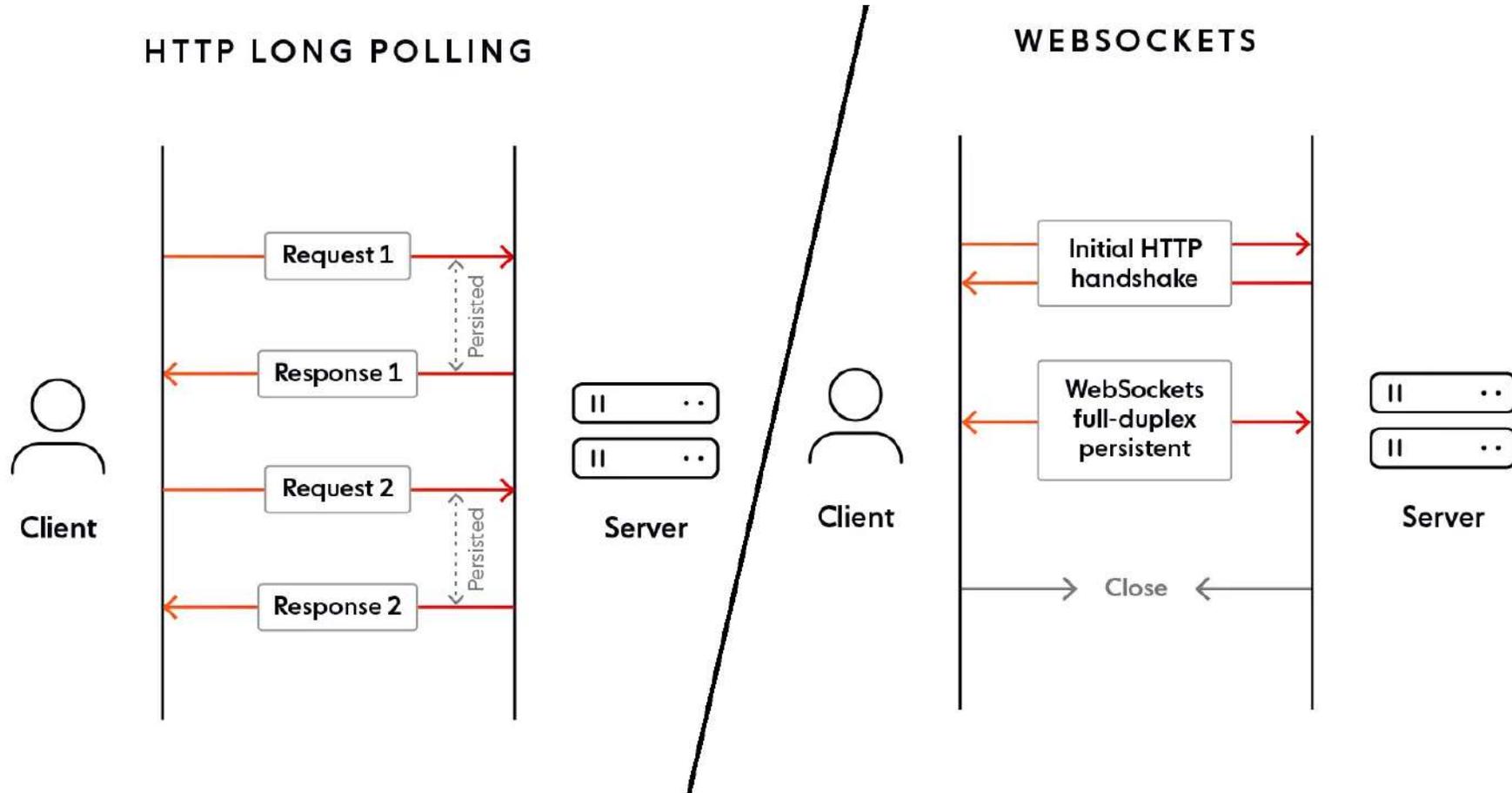
97 % hecho

GP-502	Detección de rostro	RESUELTA
GP-503	Reconocimiento del rostro identificado	RESUELTA
GP-504	Detección de posición del rostro (perfiles izquierdo y derecho, frontal)	RESUELTA
GP-505	Detección de pestaño	RESUELTA
GP-506	Identificación de emociones	RESUELTA
GP-507	Identificación de género	RESUELTA
GP-508	Estimación de edad	RESUELTA
GP-509	Estimación de raza	RESUELTA
GP-510	Extensión de funcionalidades en KeyCloak	RESUELTA
GP-511	Detección de fraude (Video en dispositivos móviles)	RESUELTA
GP-512	Challenge (Liveness Proof) con los modelos encontrados en el lado del Server	RESUELTA
GP-513	Creación de API para consumo WS y Sockets	RESUELTA
GP-514	Challenge (Liveness Proof) en el lado del Cliente con verificación en Server	RESUELTA
GP-515	Uso de la federación personalizada para el match con el userID	CERRADA
GP-516	Modificación de modelos para unificarlos mediante la transferencia de aprendizaje	CERRADA
GP-517	Transformación de modelos entrenados de servidor a cliente	RESUELTA
GP-533	Unit Test	RESUELTA
GP-677	Flujo Keycloak parametrizable	CERRADA
GP-678	Registro del faceld por parte del usuario	RESUELTA
GP-679	Registro del faceld por parte del administrador	RESUELTA

Añadir un comentario...

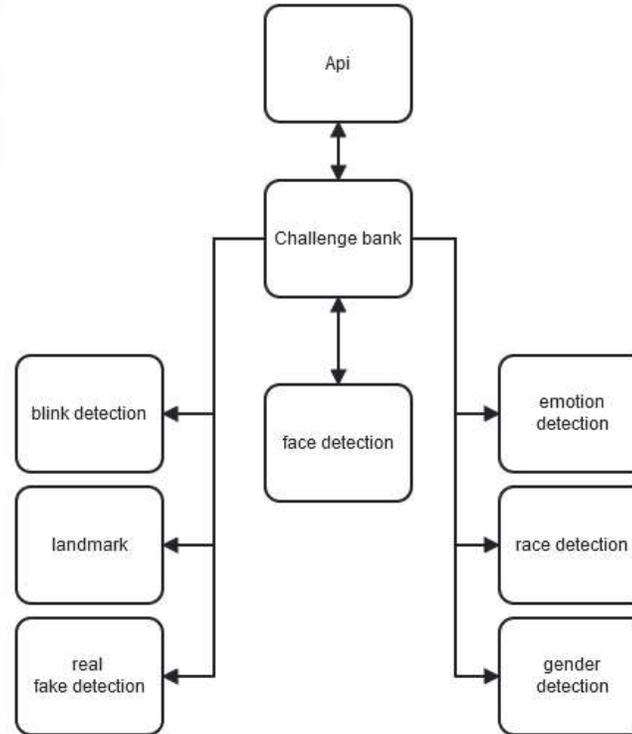
Desarrollo del sistema

# PROTOCOLO DE COMUNICACIÓN

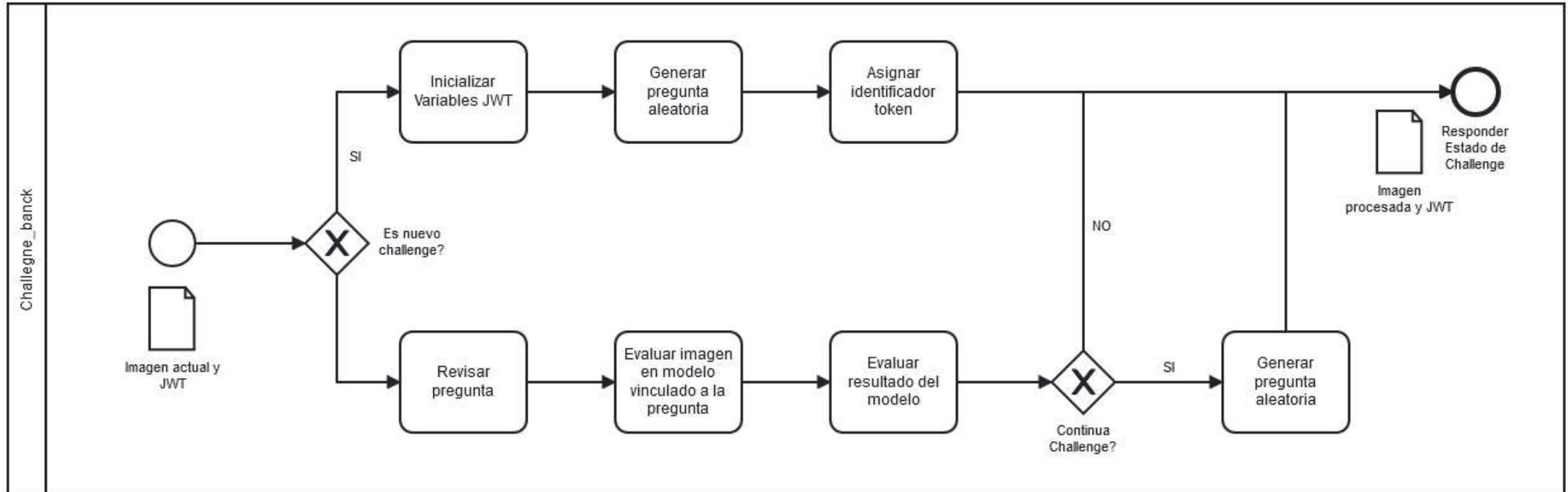


# ARQUITECTURA

```
▼ CFAVORITA-FACIAL-RECOING-CORE
  > __pycache__
  > .vscode
  > age_detection
  > api
  > ApiStorage
  > blink_detection
  > challenge_bank
  > ci
  > emotion_detection
  > face_data_recognition
  > face_verification
  > gender_detection
  > landmark
  > profile_detection
  > race_detection
  > readme-assets
  > real_fake_detection
  > unit_test
  > utils
  > .dockerignore
  > .gitignore
  > config.py
```



# BANCO DE PREGUNTAS



# EXTENDER FUNCIONALIDADES DE KEYCLOAK

## Authentication

Flows Bindings Required Actions Password Policy OTP Policy WebAuthn Policy ?  
WebAuthn Passwordless Policy ?

Browser		Requirement				New	Copy
Auth Type		REQUIRED	ALTERNATIVE	DISABLED	CONDITIONAL		
Cookie		<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>			
Kerberos		<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>			
Identity Provider Redirector		<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>			Actions
Forms		<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>		
	Username Password Form	<input checked="" type="radio"/>					
	Browser - Conditional OTP	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>		
	Condition - User Configured	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>			
	OTP Form	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>			

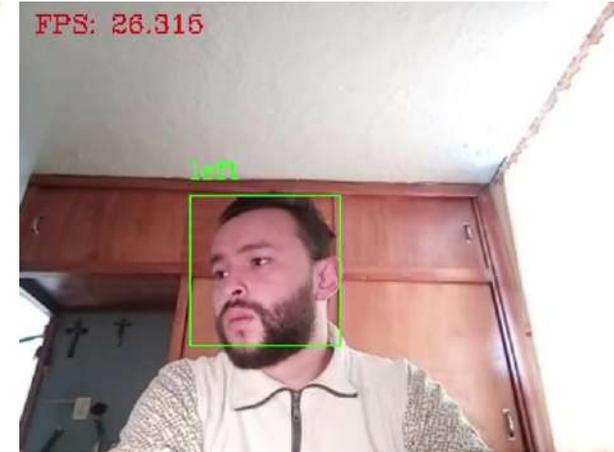
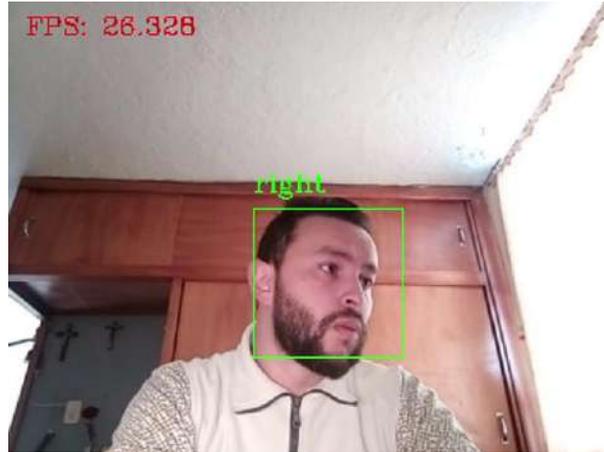
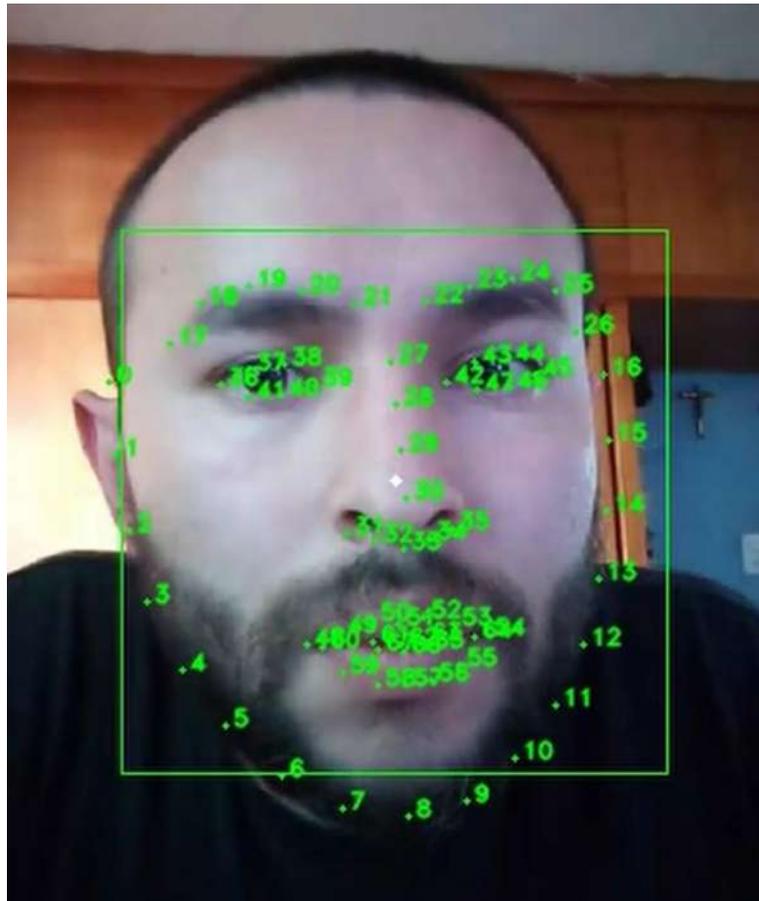
## Authentication

Flows Bindings Required Actions Password Policy OTP Policy WebAuthn Policy ?  
WebAuthn Passwordless Policy ?

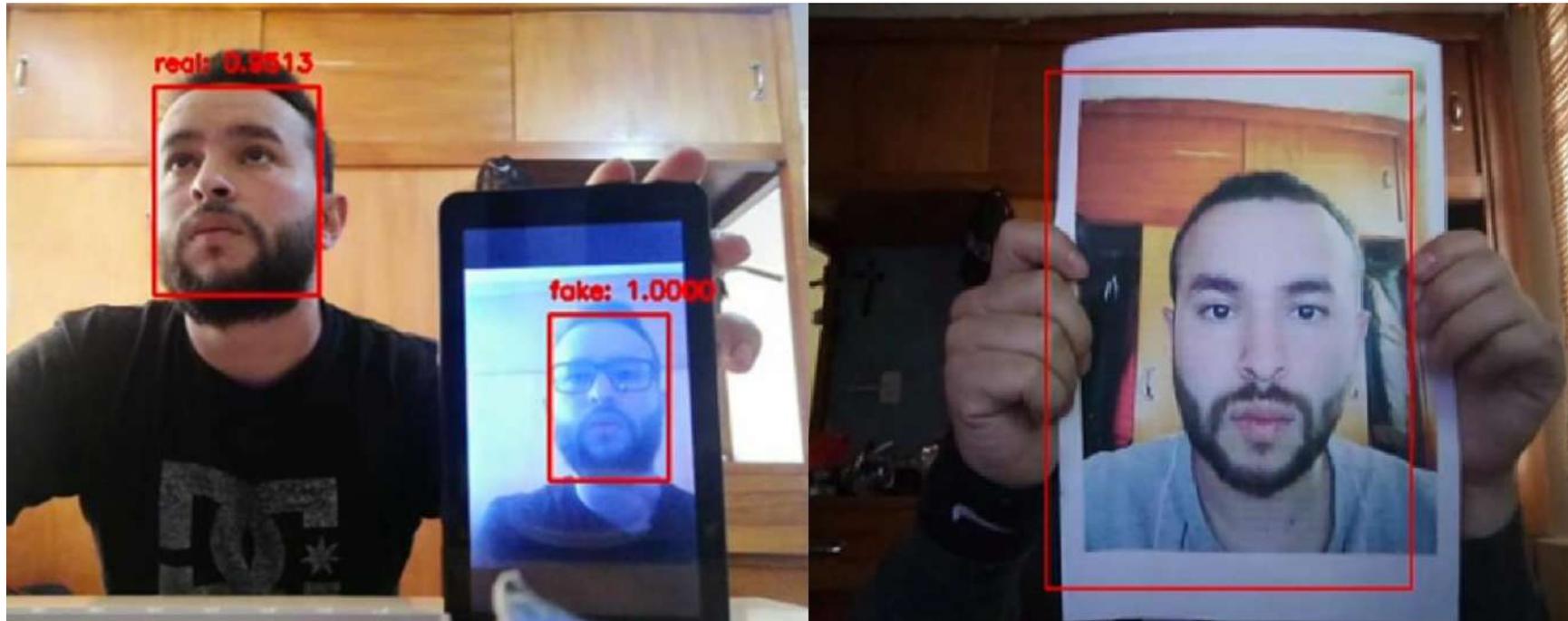
Browser Liveness Proof		Requirement				New	Copy	Delete	Edit Flow	Add execution	Add flow
Auth Type		REQUIRED	ALTERNATIVE	DISABLED	CONDITIONAL						
Cookie		<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>							Actions
Kerberos		<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>							Actions
Identity Provider Redirector		<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>							Actions
Browser Liveness Proof Forms		<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>						Actions
	Username Password Form	<input checked="" type="radio"/>									Actions
	Browser Liveness Proof Browser - Conditional OTP	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>						Actions
	Condition - User Configured	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>							Actions
	OTP Form	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>							Actions
Face On Server Side Authentication		<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>							Actions

Desarrollo del sistema

# MODELOS ACEPTADOS



# MODELOS DESCARTADOS

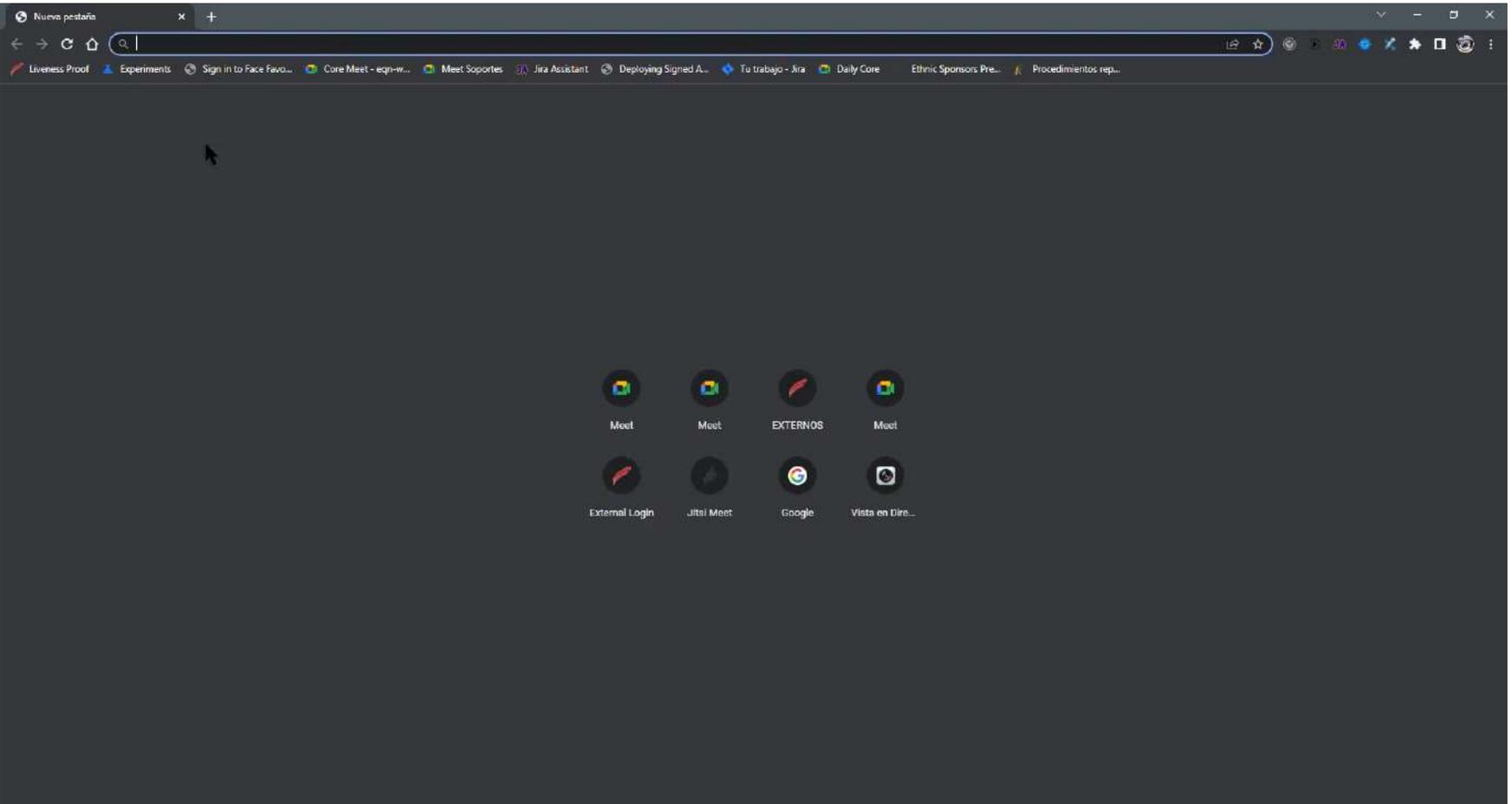


# PRUEBAS DE MODELOS



*Análisis de resultados*

# SISTEMA EN PREPRODUCCIÓN



*Análisis de resultados*

# CONCLUSIONES

No se encontró diferencia significativa entre los modelos de detección de rostro, por lo cual, la utilización de cualquiera de los mencionados en este documento puede ser sostenible para la solución propuesta.

La mejor opción fue utilizar una comunicación websocket bidireccional entre el cliente y el servidor, ya que la propuesta de solución requería un canal de comunicación en tiempo real.

La solución propuesta fue diseñada para integrarse correctamente al proveedor de identidades keycloak.

# RECOMENDACIONES

Para integrar diferentes modelos de inteligencia artificial hay que tener en cuenta la forma peculiar de utilizar cada uno de esos modelos, y estandarizar el consumo usando patrones estructurales como el Adapter que permite la colaboración entre objetos con interfaces incompatibles.

Para una efectiva ejecución de la solución propuesta en este documento se recomienda la utilización de hardware acelerado por tarjetas gráficas porque permiten la paralelización de las inferencias requeridas de diferentes modelos de inteligencia artificial.

GRACIAS POR SU ATENCIÓN

¿PREGUNTAS?

