

Detector IA



Alumno: Iván Baños Piñeiro
01/04/2025

Índice:

1. Introducción:	4
1.1. Propósito	4
1.2. Contexto de uso	4
2. Planificación del proyecto:	4
2.1. Modelo de desarrollo	4
2.2. Fases del proyecto	4
2.2.1. Análisis de requerimientos (Semana 1)	4
2.2.2. Diseño (Semanas 2-3)	4
2.2.3. Implementación (Semanas 4-7)	5
2.2.4. Pruebas (Semanas 8-9)	5
2.2.5. Despliegue (Semana 10)	5
2.2.6. Mantenimiento y mejoras (Semana 11 en adelante)	5
2.3. Diagrama de Gantt	6
3. Tecnologías y Bibliotecas Utilizadas:	7
4. Presupuesto del proyecto:	8
4.1 Coste del Hardware empleado	8
4.2 Software y herramientas empleadas	9
4.3 Total general	9
5. Diagramación:	9
5.1. Universo de la base de datos	9
5.1.1. Tablas de la base de datos	10
5.1.2. Relación entre las tablas	10
5.2. Diagrama de la ER/EER	11
5.3. Modelo relacional	12
5.4. Diagrama de clases	13
5.5. Diagramas de casos de uso	14
5.6. Diagramas de secuencia	15
5.6.1. Cliente-servidor	15
5.6.2. Profesor-servidor	16
5.6.3. Administrador-servidor	17
5.6.4. Interno del servidor	18
5.6. Diagrama de despliegue	19
6. Diseño de la interfaz de usuario:	19
6.1. Principios de diseño utilizados	19
6.2. Paleta de colores y estilo visual	20
6.3. Tipografía utilizada	20
6.4. Iconografía	20
6.5. Mockups y estructura de navegación	20
6.5.1. Index	21
6.5.2. Login	21
6.5.3. Panel de Control (administrador)	22
6.5.4. Usuarios	23

6.5.5.Editar Keywords.....	23
6.5.6.Horarios.....	24
6.5.7.Perfil del usuario.....	24
7.Validación, accesibilidad y testing.....	25
7.1.Métodos utilizados:.....	25
7.2.Resultados de la validación y mejoras aplicadas:.....	26
7.3.Conclusión:.....	26
8.Conclusiones, propuestas de mejora y valoración personal.....	26
8.1Conclusiones generales:.....	26
8.2.Problemas encontrados y soluciones:.....	27
8.3.Propuestas de mejora:.....	27
Manual de instalación.....	28
1.Requisitos técnicos.....	28
2.Instalación del software.....	28
2.1.Servidor:.....	29
2.1.1.Opción 1: Anaconda (entorno virtual).....	29
2.1.2.Opción 2: Docker (recomendable).....	32
2.2.Cliente.....	35
2.2.1.Anaconda (entorno virtual).....	35
2.2.2.Instalador (recomendable).....	37
3.Despliegue del sistema.....	38
3.1.Servidor.....	38
3.2.Cliente.....	39
Manual de usuario.....	41
1. Arquitectura general del sistema.....	41
2.Acceso y autenticación.....	41
3.Funciones según el rol del usuario.....	42
3.1.Para administradores.....	42
3.1.1.Gestión de usuarios:.....	42
3.1.2.Configuración de horarios:.....	44
3.1.3.Visualización de capturas:.....	47
3.1.4.Editar los keywords:.....	50
3.2.Para profesores.....	50
4.Cierre de sesión.....	50
Bibliografía.....	51

1. Introducción:

1.1. Propósito

Este proyecto tiene como finalidad desarrollar un sistema integral para monitorear el uso de inteligencia artificial (IA) por parte de estudiantes durante exámenes o pruebas académicas, detectando actividades no permitidas mediante capturas de pantalla y procesamiento de texto en tiempo real.

1.2. Contexto de uso

El objetivo principal del sistema es garantizar la integridad académica mediante la detección de posibles fraudes, utilizando técnicas avanzadas de captura y procesamiento de texto automatizado.

Los usuarios principales del sistema son:

Administradores: encargados de gestionar usuarios, horarios y configuraciones generales del sistema.

Profesores: responsables de supervisar exámenes y revisar las alertas generadas por el sistema.

Estudiantes: usuarios finales cuyos equipos son monitorizados durante evaluaciones.

2. Planificación del proyecto

2.1. Modelo de desarrollo

Para llevar a cabo este proyecto se ha seguido un modelo de desarrollo en cascada que permite una organización clara y efectiva. Este modelo se divide en varias fases secuenciales donde cada fase depende del resultado de la anterior.

2.2. Fases del proyecto

2.2.1. Análisis de requerimientos (Semana 1)

- Identificación detallada de requisitos funcionales y no funcionales.
- Determinación de tecnologías a emplear.

2.2.2. Diseño (Semanas 2-3)

- Diagramas de clases y modelo relacional.
- Diseño de base de datos y diagramas EER.
- Diseño de interfaz de usuario mediante mockups.

2.2.3.Implementación (Semanas 4-7)

- Desarrollo del backend con Flask.
- Desarrollo del cliente Python con PyAutoGUI y Tesseract OCR.
- Creación y ajuste del frontend usando Bootstrap.

2.2.4.Pruebas (Semanas 8-9)

- Pruebas unitarias, de integración y de sistema.
- Validación de accesibilidad y usabilidad.

2.2.5.Despliegue (Semana 10)

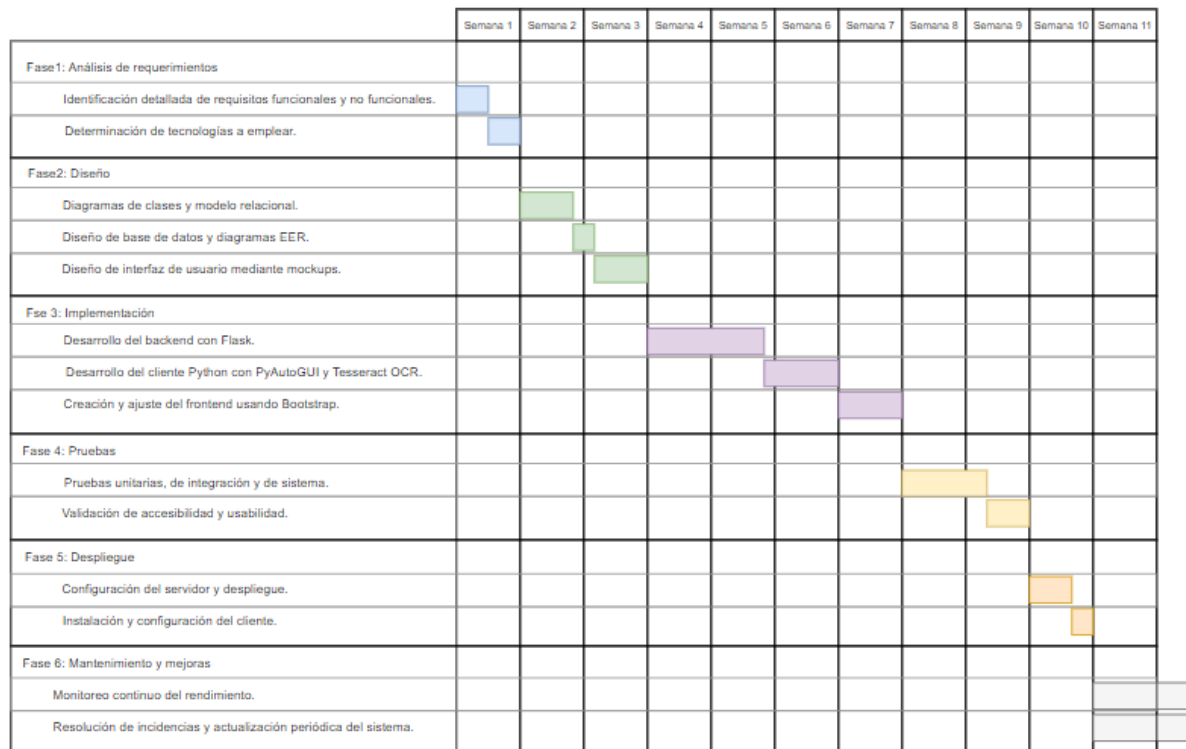
- Configuración del servidor y despliegue.
- Instalación y configuración del cliente.

2.2.6.Mantenimiento y mejoras (Semana 11 en adelante)

- Monitoreo continuo del rendimiento.
- Resolución de incidencias y actualización periódica del sistema.

Esta planificación asegura un desarrollo ordenado, eficiente y con alta calidad del producto final.

2.3.Diagrama de Gantt



3.Tecnologías y Bibliotecas Utilizadas

- **Flask** (Backend en Python)
 - Microframework web en Python.
 - Permite definir rutas, manejar peticiones y renderizar plantillas HTML.
 - Se usa `render_template`, `request`, `Blueprint`, etc.
- **Flask-Login** (Autenticación)
 - Extensión de Flask para gestionar sesiones de usuario.
 - Funciones clave: `login_user()`, `logout_user()`, `@login_required`, `current_user`.
- **Flask-SQLAlchemy** (ORM)
 - ORM para bases de datos SQL con integración en Flask.
 - Permite interactuar con la base de datos sin escribir SQL manualmente.
 - Define los modelos: Usuario, Equipo, Datos, Horario.
- **Blueprints en Flask**
 - Permite dividir la aplicación en módulos independientes.
- **SQLite** (Base de Datos)
 - Motor de base de datos embebido que guarda toda la información en un archivo `.db`.
 - Es simple y eficaz para proyectos pequeños, pero tiene limitaciones con múltiples escrituras concurrentes.
- **SQLAlchemy.exc** (Gestión de Errores de BD)
 - Módulo de excepciones de SQLAlchemy.
 - Se utiliza para detectar fallos en la base de datos como colapsos, errores de escritura o corrupción.
 - Permite capturar excepciones como `OperationalError`, `IntegrityError`, etc.
- **queue** (Sistema de Colas)
 - Biblioteca estándar de Python para crear colas FIFO.
 - En tu caso, se usa para gestionar múltiples escrituras a la base de datos de forma secuencial y evitar bloqueos por concurrencia.
- **Werkzeug** (Seguridad)
 - Usado para `generate_password_hash()` y `check_password_hash()`, protegiendo las contraseñas.
- **ConfigParser**
 - Permite leer archivos `.ini` con configuraciones del cliente o servidor.
- **PIL** (Pillow)
 - Biblioteca de procesamiento de imágenes.
 - Usada para manipular y preparar las capturas antes de enviarlas al servidor.
- **PyAutoGUI**
 - Permite automatizar capturas de pantalla en el cliente.
- **Pytesseract**
 - OCR basado en Tesseract.
 - Extrae texto de las imágenes para detectar palabras clave.
- **PyGetWindow**
 - Detecta la ventana activa en el sistema operativo.
 - Ayuda a identificar cuándo el usuario cambia de aplicación.

- **Requests**
 - Librería HTTP para Python.
 - Utilizada por el cliente para comunicarse con el servidor (GET /keywords, POST /uploads).
- **Pickle**
 - Serializa datos Python (como listas de palabras clave) en archivos .pkl.
- **os**
 - Interacción con el sistema operativo: rutas, nombres de equipos, variables de entorno.
- **datetime y time**
 - Manejo de fechas, tiempos, y temporizadores (time.sleep(1) para ciclos de monitoreo).
- **HTML** (Frontend)
 - Lenguaje de marcado para estructurar contenido web.
 - Utilizado junto con Jinja2 para mostrar datos dinámicamente.
- **CSS**
 - Lenguaje de estilos para definir el diseño visual de las páginas web.
- **JavaScript**
 - Lenguaje de programación para el navegador.
 - Aporta dinamismo e interactividad al frontend.
- **AJAX** (con JS Fetch o jQuery)
 - Peticiones HTTP asíncronas desde el cliente al servidor, sin recargar la página.
 - Arquitectura y Escalabilidad

4.Presupuesto del proyecto:

4.1 Coste del Hardware empleado

Componente	Precio
Ordenador completo (procesador Intel i7, 16GB RAM, 1TB SSD)	950 €
Pantalla Full HD 24"	150 €
Periféricos (teclado y ratón)	50 €
Subtotal Hardware	1.150 €

4.2 Software y herramientas empleadas

Componente	Precio
Sistema Operativo (Windows 10)	110,99 €
Python 3.8.20	Gratuito
Anaconda (Gestión de entorno Python)	Gratuito
Google docs	Gratuito
Visual Studio Code (Editor de código)	Gratuito
GitHub (Control de versiones)	Gratuito
Draw.io	Gratuito
Google Chrome	Gratuito
Subtotal Software	110,99€

4.3 Total general

Concepto	Precio
Subtotal Hardware	1.150 €
Subtotal Software	110,99 €
Total General	1.250,99 €

Este presupuesto refleja un proyecto asequible basado principalmente en soluciones gratuitas y open-source, optimizando los costes totales.

5.Diagramación

5.1. Universo de la base de datos

Este programa está pensado para detectar cuando los estudiantes utilizan inteligencia artificial durante exámenes o clases normales. Para ello, necesitamos almacenar información sobre quién puede acceder al sistema, los horarios en los que el sistema debe vigilar los ordenadores, y las capturas de pantalla que se realizan automáticamente cuando se detecta algo sospechoso. Además, tenemos que registrar qué profesor está relacionado con cada captura, permitiéndole acceder únicamente a sus propias imágenes y controlar de forma sencilla cuándo activar o desactivar el modo automático de capturas según la necesidad del momento.

5.1.1.Tablas de la base de datos:

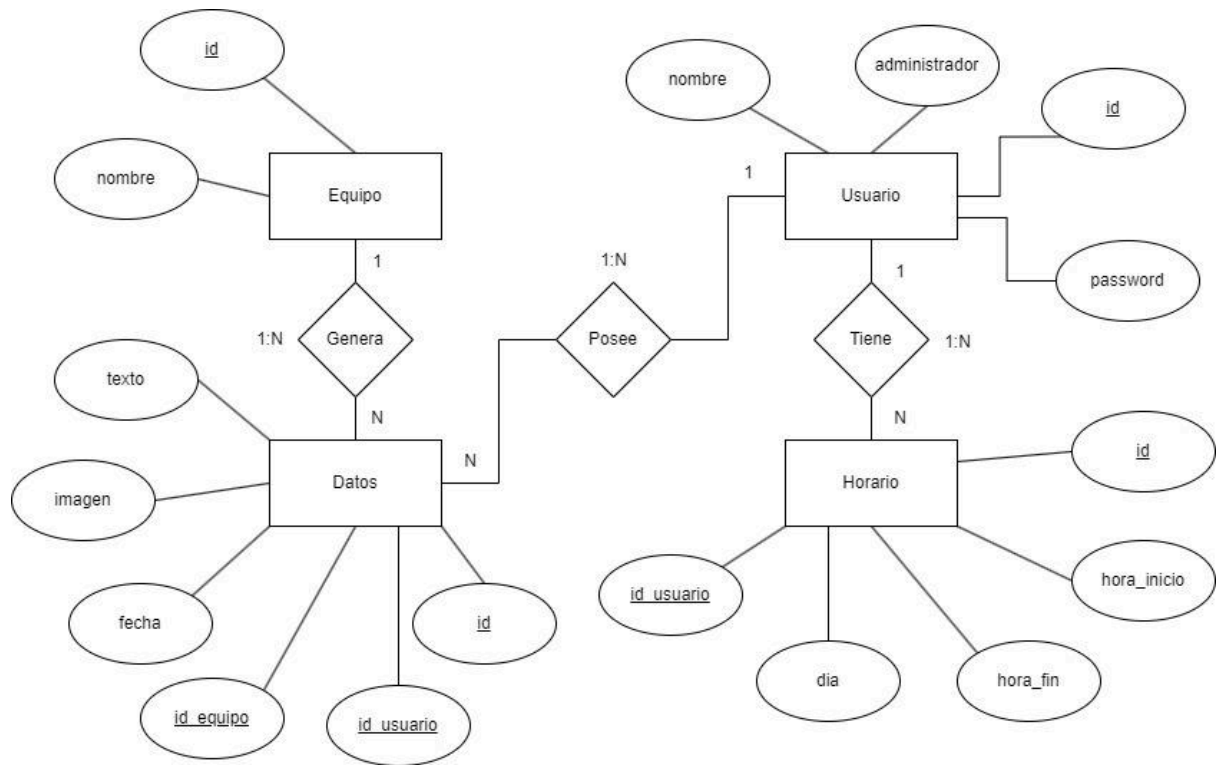
- **Tabla usuarios**
 - id (int, PK) → Identificador único del usuario.
 - nombre (str, único) → Nombre de usuario.
 - password_hash (str) → Hash de la contraseña.
 - administrador (bool) → Indica si el usuario tiene permisos de administrador.
- **Tabla equipos**
 - id (int, PK) → Identificador único del equipo.
 - nombre (str, único) → Nombre del equipo.
- **Tabla datos**
 - id (int, PK) → Identificador único de la captura.
 - id_usuario (int, FK) → Usuario propietario de la captura.
 - id_equipo (int, FK) → Equipo que generó la captura.
 - fecha (datetime) → Fecha y hora de la captura.
 - imagen (LargeBinary) → Imagen capturada.
 - texto (Text) → Texto extraído mediante OCR.
- **Tabla horarios**
 - id (int, PK) → Identificador único del horario.
 - hora_inicio (Time) → Hora de inicio de la captura.
 - hora_fin (Time) → Hora de finalización de la captura.
 - dia (int) → Día de la semana (0=Lunes, 6=Domingo).
 - id_usuario (int, FK) → Usuario asociado al horario.

5.1.2.Relación entre las tablas

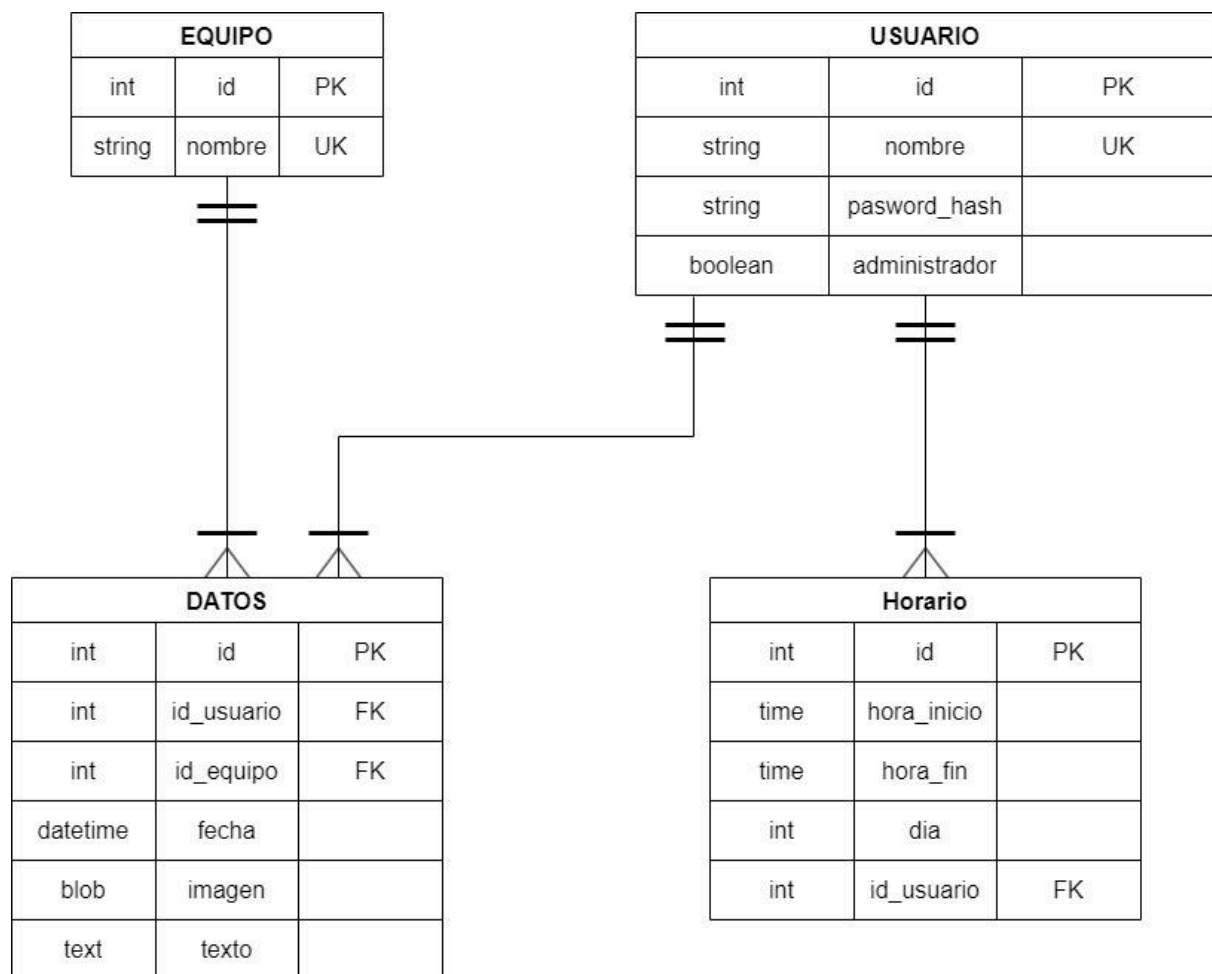
- Un usuario puede tener múltiples horarios asignados. Esto permite definir diferentes tramos de tiempo en los que ese usuario (normalmente un profesor) tiene clases y desea activar el sistema de capturas.
- Un usuario también puede estar asociado a múltiples datos (capturas), de forma que cada imagen enviada por los clientes quede registrada con el profesor correspondiente.
- Cada equipo puede enviar múltiples datos. Esto permite que desde un mismo ordenador se registren todas las capturas realizadas durante diferentes sesiones.
- La tabla datos sirve de nexo entre usuarios y equipos, reflejando qué profesor estaba a cargo en el momento de la captura y desde qué dispositivo se tomó.

Esta estructura de base de datos permite una gestión eficiente del monitoreo de IA en los exámenes, asegurando que cada usuario solo acceda a la información que le corresponde y garantizando la integridad del sistema.

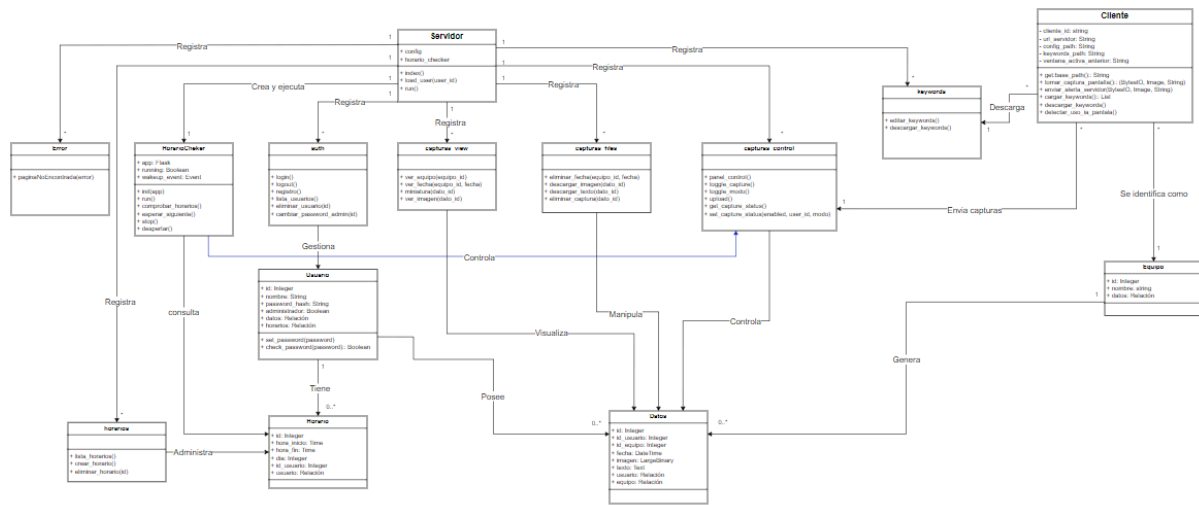
5.2. Diagrama de la ER/EER



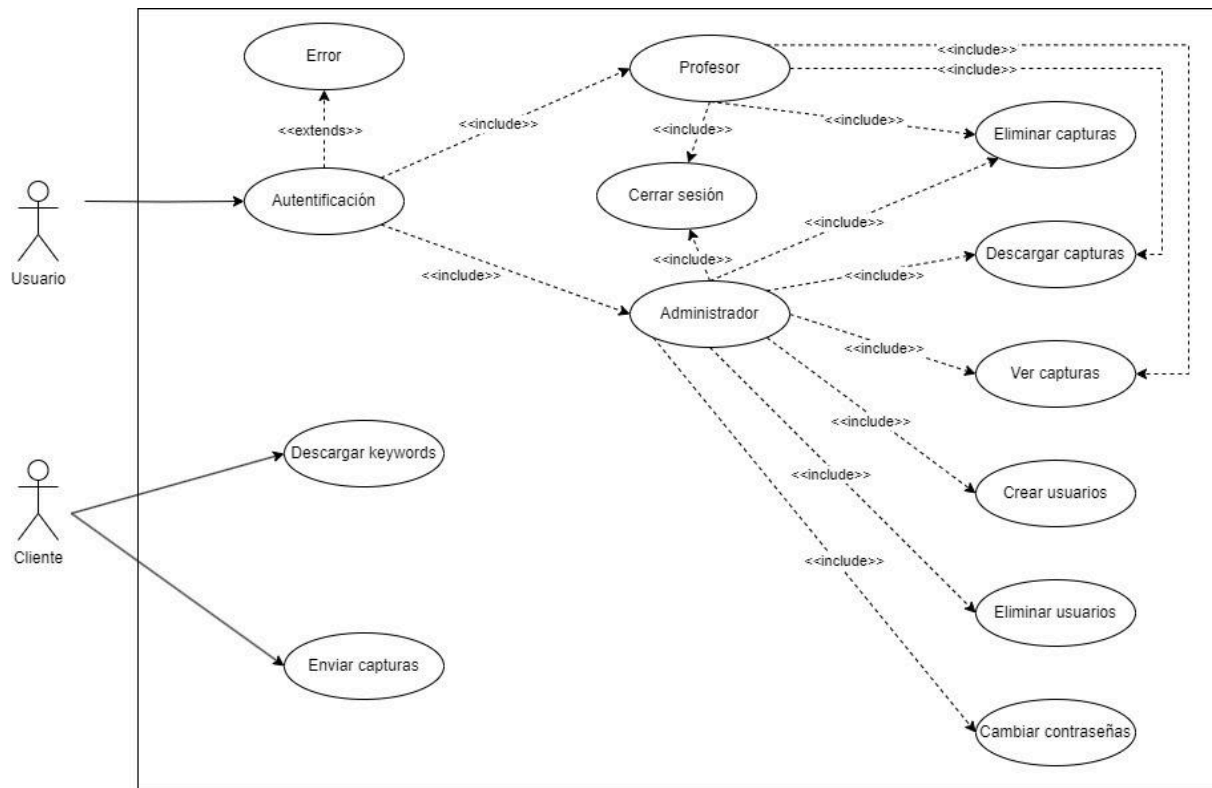
5.3.Modelo relacional



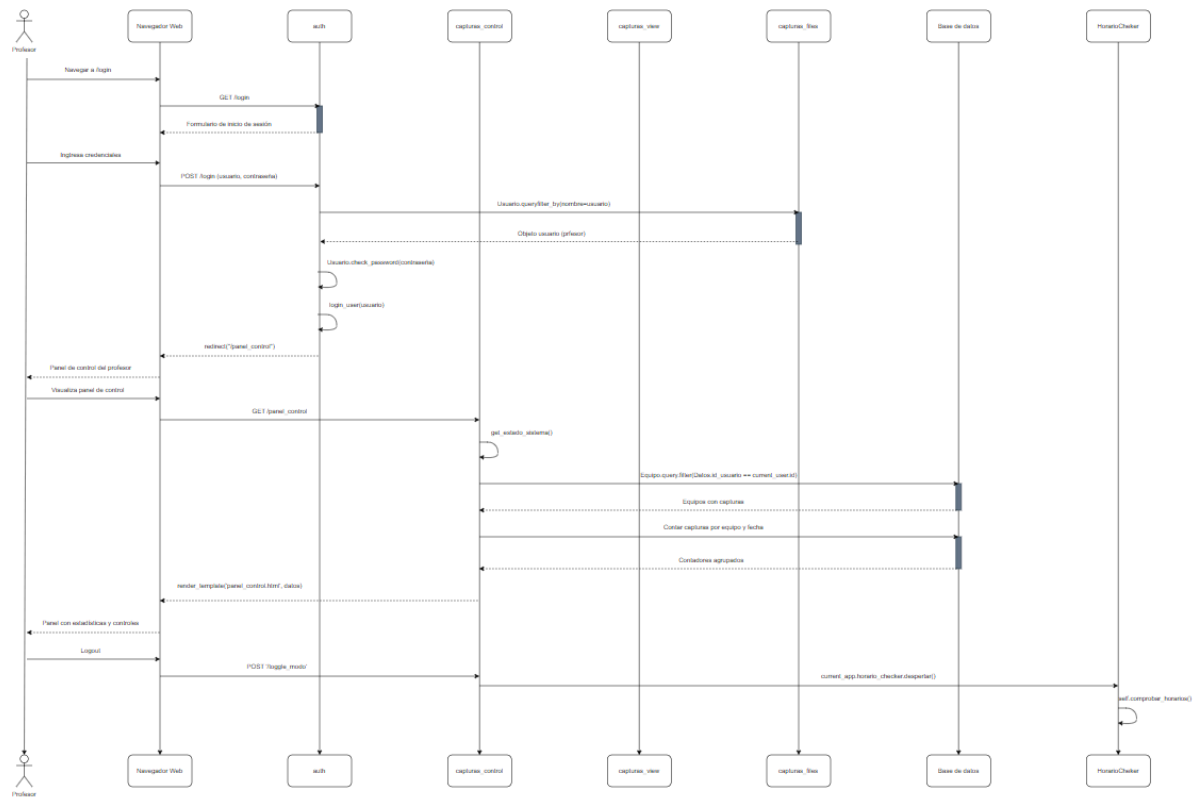
5.4. Diagrama de clases



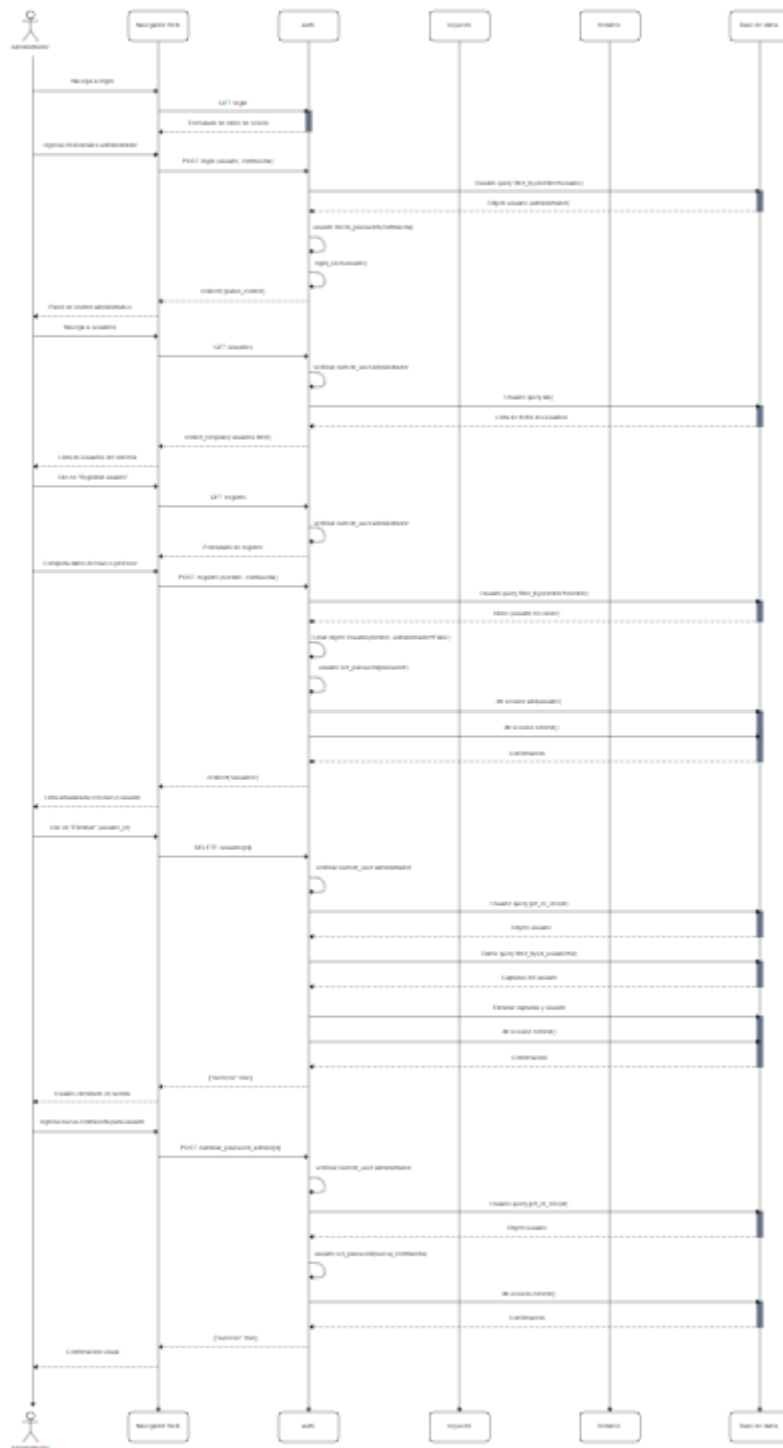
5.5. Diagramas de casos de uso



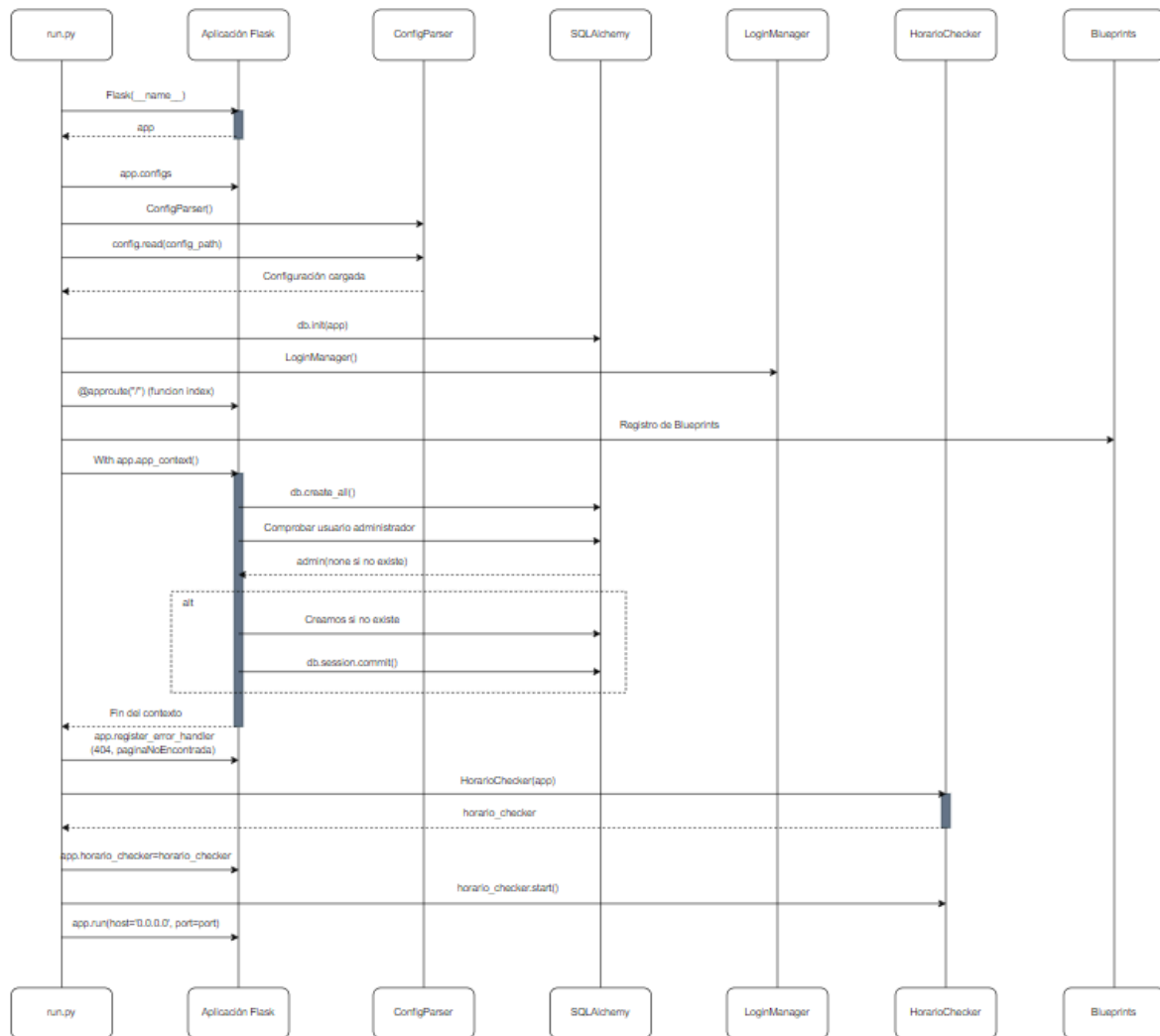
5.6.2. Profesor-servidor



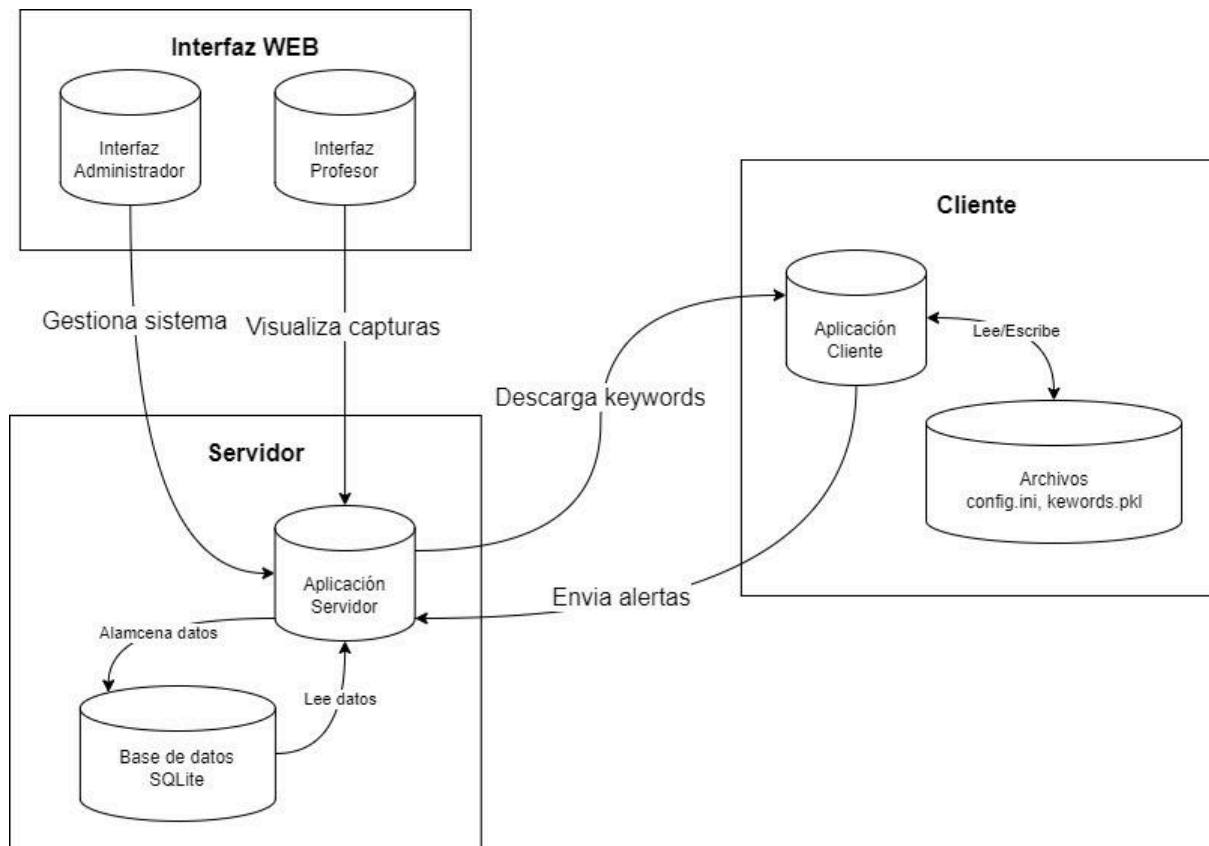
17



5.6.4. Interno del servidor



5.6. Diagrama de despliegue



6. Diseño de la interfaz de usuario

La interfaz de usuario del sistema ha sido diseñada con el objetivo de ser clara, intuitiva y eficiente para cada uno de los roles que interactúan con la aplicación. Se ha utilizado un enfoque basado en Bootstrap para garantizar un diseño responsive que se adapte a distintos dispositivos, facilitando la accesibilidad tanto en ordenadores como en tablets o móviles.

6.1. Principios de diseño utilizados:

- **Simplicidad:** La interfaz minimiza distracciones y presenta únicamente la información relevante.
- **Consistencia:** Se mantiene un estilo homogéneo en todas las vistas.
- **Accesibilidad:** Se han aplicado buenas prácticas para asegurar la usabilidad por parte de todos los usuarios.
- **Optimización de flujo de trabajo:** Cada acción importante se puede realizar en pocos pasos.

6.2. Paleta de colores y estilo visual:

Se ha optado por una paleta de colores sobria y profesional, combinando tonos oscuros y neutros para facilitar la lectura prolongada y reducir la fatiga visual. Los colores clave son:

- Negro (#212529): Color predominante para encabezados y elementos de énfasis.
- Gris claro (#F7F7F7): Fondo de las secciones principales para crear contraste.
- Blanco (#FFFFFF): Utilizado en textos y fondos secundarios.
- Verde (#10B981): Indica acciones exitosas y estados positivos.
- Rojo (#EF4444): Se usa para alertas y acciones críticas, como eliminación de capturas.
- Azul (#0B5ED7): Para los botones de descargar imágenes, y textos (links)
- Azul claro (#0DCAF0): Para los botones de descargar texto
- Gris (#C7C8AF): Usado para textos del navBar

6.3. Tipografía utilizada:

El sistema emplea la fuente Inter, que proporciona una lectura clara y profesional en distintos tamaños. Se han definido los siguientes tamaños para mejorar la jerarquía visual:

- Títulos principales: 24px, negrita.
- Subtítulos: 18px, seminegrita.
- Texto normal: 16px.
- Etiquetas y botones: 14px.

6.4. Iconografía:


Se han utilizado imágenes obtenidas de Google y un icono de aplicación generado con DALL·E.

6.5. Mockups y estructura de navegación:

El sistema cuenta con distintas pantallas diseñadas para la gestión eficiente de sus funciones. Las principales vistas implementadas son:

6.5.1.Index

Página de inicio del sistema.

 IA Detector Login

Bienvenido a IA Detector


Sistema de detección y monitoreo de actividad sospechosa


Inicia sesión para acceder al sistema

Iniciar sesión

6.5.2.Login

Permite entrar en un usuario.

 IA Detector Login



Inicio de sesión

Usuario

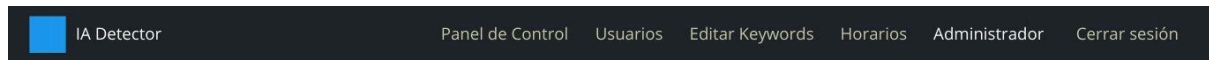
Contraseña

Iniciar sesión

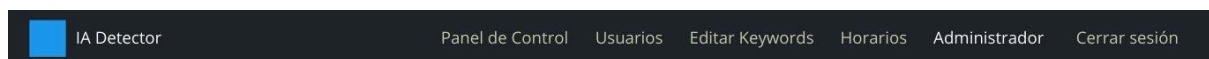
(C) 2025

6.5.3. Panel de Control (administrador)

Sección principal para la gestión de capturas y estado del sistema.



[Panel de control](#) / [PC-1](#)



[Panel de control](#) / [PC-1](#) / [2025-03-21](#)



6.5.4.Usuarios

Administración de usuarios, accesible solo para administradores.

Lista de Usuarios

Nuevo Usuario

ID	Nombre	Administrador	Acciones	
1	Administrador	Si	Cambiar Contraseña	Eliminar
2	Usuario 1	No	Cambiar Contraseña	Eliminar
3	Usuario 2	No	Cambiar Contraseña	Eliminar
4	Usuario 3	No	Cambiar Contraseña	Eliminar

6.5.5.Editar Keywords

Gestión de palabras clave para detección de IA.

Editar Palabras Clave

Descargar Archivo

Contenido del archivo keywords.txt

chatgpt
copilot
openai
bard
codeium
qwen
deepseek
davinci
codex
watson
lumen5
hivemind
chatbot
deepmind

Guardar Cambios

6.5.6.Horarios

Configuración de los períodos en los que se activará la monitorización.

IA Detector

Panel de Control

Usuarios

Editar Keywords

Horarios

Administrador

Cerrar sesión

Gestión de Horarios

Añadir Nuevo Horario

Usuario

Día

Hora inicio

Hora fin

Seleccionar usuario ▼

Lunes ▼

--:--

--:--

Añadir Horario

Lunes

Martes

Miercoles

Jueves

Viernes

Sabado

Domingo

Usuario	Hora inicio	Hora fin	Acciones
Usuario 1	18:20	19:20	Eliminar
Usuario 2	19:20	20:20	Eliminar

6.5.7.Perfil del usuario

Muestra el nombre del usuario activo y opciones de personalización.

IA Detector

Panel de Control

Usuario 1

Cerrar sesión

Panel de control

Control de Capturas

☐

Modo Automático

Estado del Sistema: Desactivado

Iniciar Capturas

Capturas por Equipo

E

PC-1

5 días

E

PC-2

7 días

E

PC-3

1 días

E

PC-4

15 días

E

PC-5

8 días

E

PC-6

12 días

Panel de control

Control de Capturas

☐
Modo Automático

Estado del Sistema: Activado

Guardando para: Usuario 1 (Modo Manual)

Detener captura

Capturas por Equipo

E	E	E	E	E	E
PC-1	PC-2	PC-3	PC-4	PC-5	PC-6
5 días	7 días	1 días	15 días	8 días	12 días

Este diseño garantiza una experiencia de usuario optimizada, permitiendo que administradores y profesores gestionen el sistema de manera eficiente y sin complicaciones innecesarias.

7.Validación, accesibilidad y testing

Para garantizar que el sistema sea accesible, estable y funcional en diferentes entornos, se realizaron diversas pruebas manuales enfocadas en la estabilidad, usabilidad y rendimiento del sistema.

7.1.Métodos utilizados:

- Pruebas en hardware variado: Se ejecutó el sistema en un ordenador moderno y en un portátil con especificaciones limitadas para comprobar su rendimiento y fluidez.
- Evaluación visual manual: Se revisó la claridad de la interfaz, la legibilidad de los textos y la facilidad de navegación.
- Simulación de distintos escenarios de uso: Se probaron todas las funciones principales, incluyendo la activación y desactivación de capturas, la gestión de horarios y la administración de palabras clave.
- Pruebas funcionales: Se verificó que cada módulo del sistema ejecuta correctamente su función asignada, incluyendo la captura de pantalla, el análisis OCR y el envío de datos al servidor.
- Pruebas de integración: Se comprobó la interacción entre los distintos componentes del sistema, asegurando que la comunicación entre cliente y servidor se realiza sin errores.

- Pruebas de estrés: Se ejecutaron pruebas con múltiples clientes enviando capturas simultáneamente para evaluar la capacidad del servidor.
- Pruebas de compatibilidad: Se probó el sistema en diferentes dispositivos y resoluciones de pantalla para garantizar una correcta visualización.

7.2.Resultados de la validación y mejoras aplicadas:

Las pruebas realizadas confirmaron que el sistema funciona de manera estable en distintos entornos, aunque se realizaron algunos ajustes para optimizar su rendimiento:

- Optimización de capturas: Se mejoró el procesamiento de imágenes para reducir el tiempo de análisis.
- Corrección de errores en la interfaz: Se ajustaron ciertos elementos visuales que no se mostraban correctamente en resoluciones más bajas.
- Mejor manejo de datos en la base de datos: Se optimizó la consulta y almacenamiento de información para mejorar la velocidad de respuesta del servidor.
- Ajuste en el tamaño de ciertos elementos: Para mejorar su visibilidad en pantallas más pequeñas.
- Optimización de la carga de la interfaz: Para mejorar el rendimiento en equipos con menor capacidad.
- Verificación del acceso y flujo de trabajo: Para garantizar que los usuarios pueden realizar sus acciones sin inconvenientes.

7.3.Conclusión:

Gracias a estas validaciones y ajustes, el sistema se ha optimizado para ofrecer una experiencia estable y accesible en distintos dispositivos y condiciones de uso, permitiendo que administradores y profesores interactúen con la aplicación sin dificultades técnicas.

8.Conclusiones, propuestas de mejora y valoración personal

Después de todo el proceso de desarrollo, implementación y pruebas, se han obtenido diversas conclusiones que reflejan el desempeño del sistema y las posibles mejoras futuras.

8.1Conclusiones generales:

- El sistema cumple con su propósito principal de detectar el uso de inteligencia artificial en exámenes y clases, brindando una herramienta útil para administradores y profesores.
- La elección de Flask como framework backend ha permitido un desarrollo ágil y una estructura modular fácil de mantener.
- La integración de PyAutoGUI y Tesseract OCR ha demostrado ser efectiva en la captura y análisis de texto en pantalla, aunque con ciertas limitaciones en condiciones de baja calidad de imagen.

- El almacenamiento en SQLite ha resultado suficiente para la escala del proyecto, pero podría requerir una migración a una base de datos más robusta en caso de un crecimiento significativo.

8.2.Problemas encontrados y soluciones:

- Optimización del procesamiento OCR: Inicialmente, la detección de texto tenía fallos con ciertas tipografías y tamaños. Se soluciona ajustando la preprocesación de las imágenes antes del análisis.
- Fluidez en la interfaz de usuario: Algunas pruebas iniciales indicaron tiempos de carga altos en dispositivos con menor capacidad. Se resolvió optimizando consultas y reduciendo el peso de ciertos recursos gráficos.
- Gestión de horarios: Se encontró necesario mejorar la flexibilidad del sistema de horarios para permitir más personalización, lo cual fue implementado en una actualización posterior.
- Conflictos de escritura simultánea en base de datos: Uno de los problemas más críticos surgió al recibir múltiples capturas de diferentes clientes de forma simultánea. Esto provocaba errores de escritura en la base de datos SQLite, que no está diseñada para manejar accesos concurrentes intensivos. La solución implementada fue incorporar una cola de tipo Queue, que serializa las operaciones de escritura. De esta forma, las capturas se procesan una a una en segundo plano, evitando bloqueos y garantizando la integridad de los datos. Esta mejora ha permitido mantener el uso de SQLite como base de datos ligera y eficiente, incluso con múltiples clientes activos.

8.3.Propuestas de mejora:

- Migración a una base de datos más escalable: En un futuro, se podría utilizar PostgreSQL o MySQL para manejar grandes volúmenes de datos con mayor eficiencia.
- Implementación de WebSockets: Para mejorar la experiencia de los profesores, se podría desarrollar un sistema de notificaciones en tiempo real para nuevas capturas detectadas.
- Visualización de equipos conectados: Añadir una sección específica en el panel de administrador que muestre los equipos conectados recientemente o actualmente activos. Esto facilita el seguimiento en tiempo real, la detección de problemas de red y la confirmación de que los clientes están funcionando correctamente.

Manual de instalación

1.Requisitos técnicos

Requisitos mínimos recomendados:

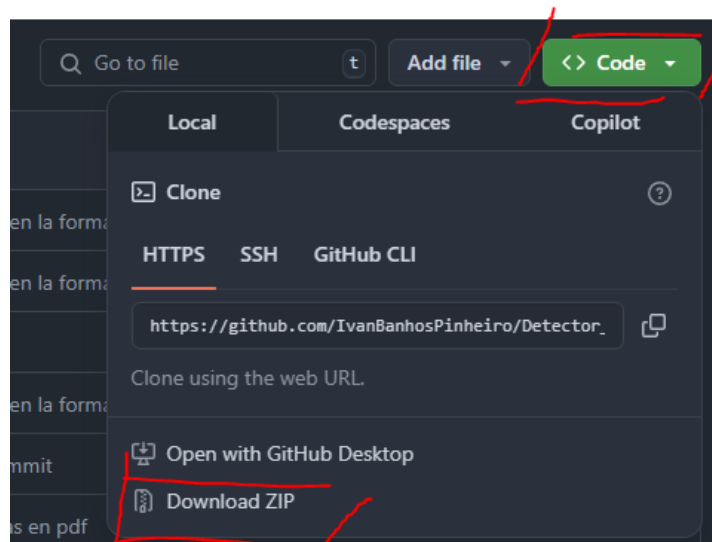
- Servidor:
 - Procesador: Intel Core i5 o superior.
 - Memoria RAM: 8GB.
 - Almacenamiento: 50GB de espacio libre.
 - Sistema Operativo: Windows 10 o superior.
 - Docker instalado para la gestión del servidor.
- Cliente (ordenador de aula):
 - Procesador: Intel Core i3 o superior.
 - Memoria RAM: 4GB.
 - Sistema Operativo: Windows 10 o superior.
 - Conexión a Internet local para comunicación con el servidor.

2.Instalación del software

Este manual describe los pasos necesarios para instalar y configurar correctamente el sistema, tanto en el servidor como en los equipos cliente.

Para ambos puntos necesitaremos descargar el proyecto de github, para eso nos dirigiremos a este link y lo descargaremos.

https://github.com/IvanBanhosPinheiro/Detector_la_Trabajo_Final



Para el cliente también descargamos el ejecutable.

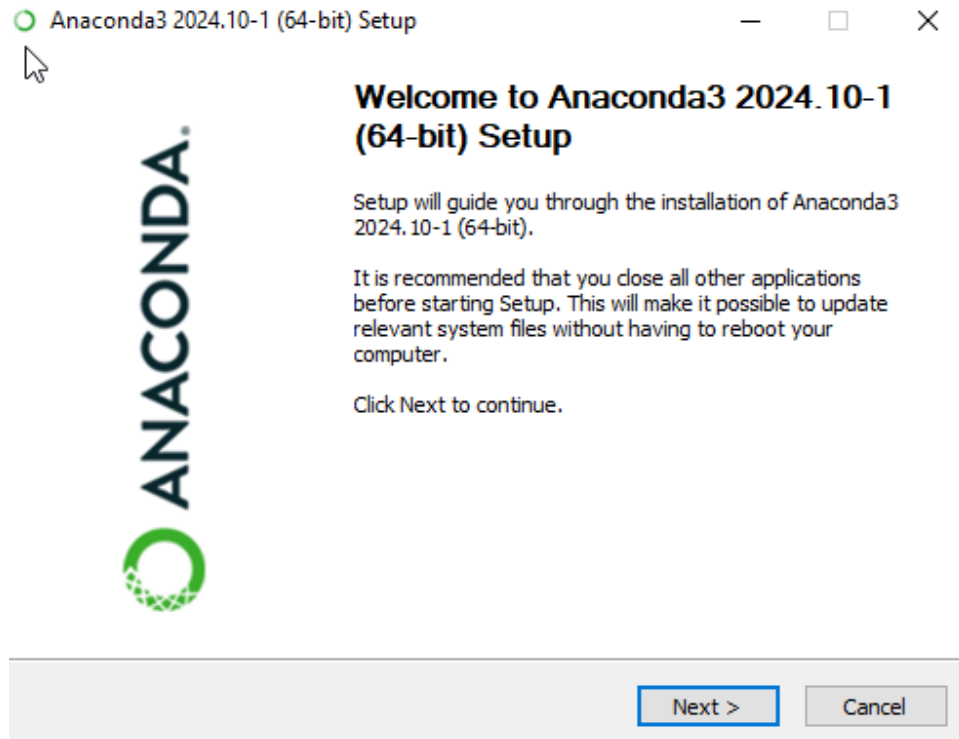
https://github.com/IvanBanhosPinheiro/Detector_la_Trabajo_Final/releases/tag/v0.1

2.1.Servidor:

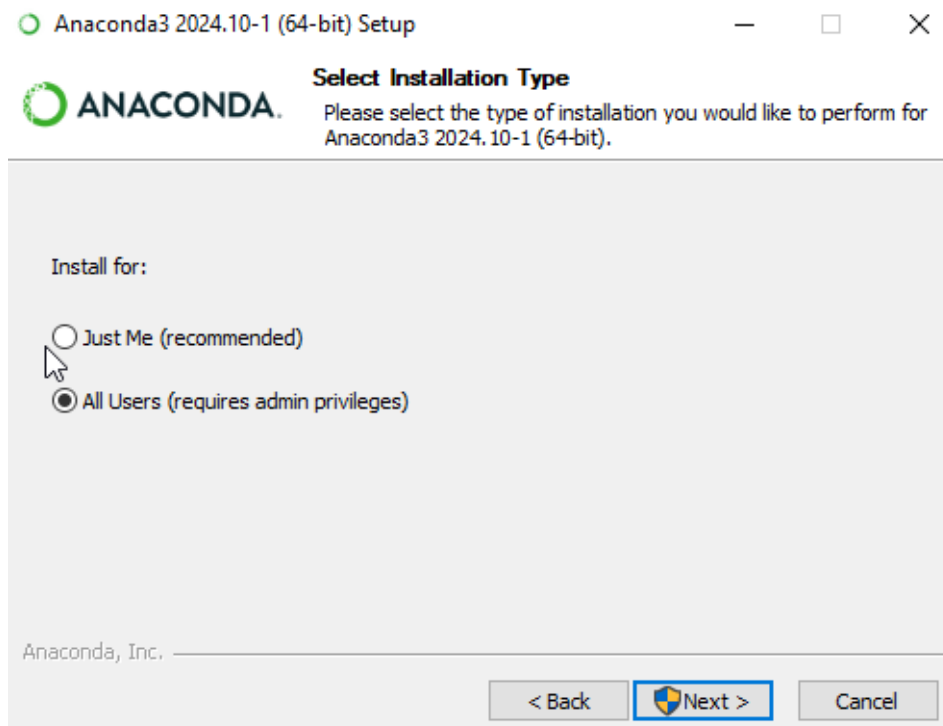
2.1.1.Opción 1: Anaconda (entorno virtual).

Descargamos anaconda <https://www.anaconda.com/download/success>

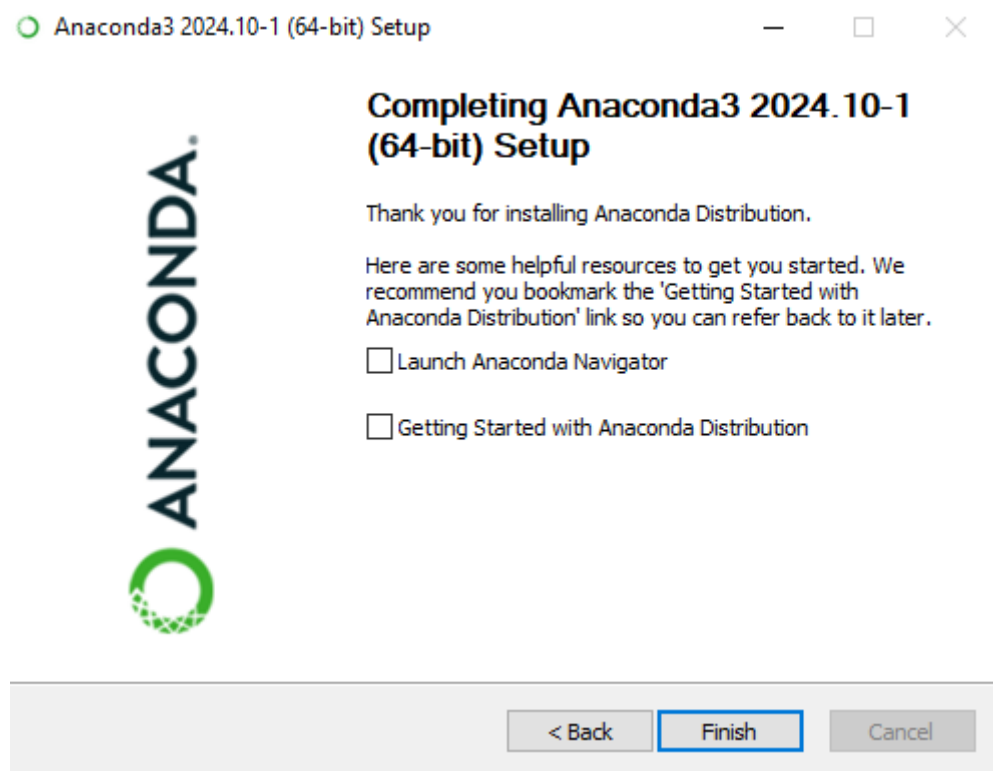
- Seguimos las instrucciones de instalación.



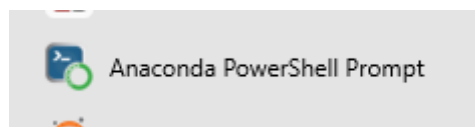
Yo seleccioné la opción de para todos que pedirá permisos de administrador



Siguiente siguiente... y finish



- Ahora vamos a abrir la consola de conda para crear un entorno virtual con python 3.8.20, en inicio



- Escribimos este comando: `conda create -n detector_env python=3.8.20`

Anaconda PowerShell Prompt

```
(base) PS C:\Users\Tutorial> conda create -n detector_env python=3.8.20_
```

Esto descargara todos los paquetes necesarios para crear un entorno con python 3.8.20 para poder usar nuestro programa.

- Una vez creado necesitamos activarlo con el siguiente comando: `conda activate detector_env` lo ejecutamos en la consola de conda tambien.

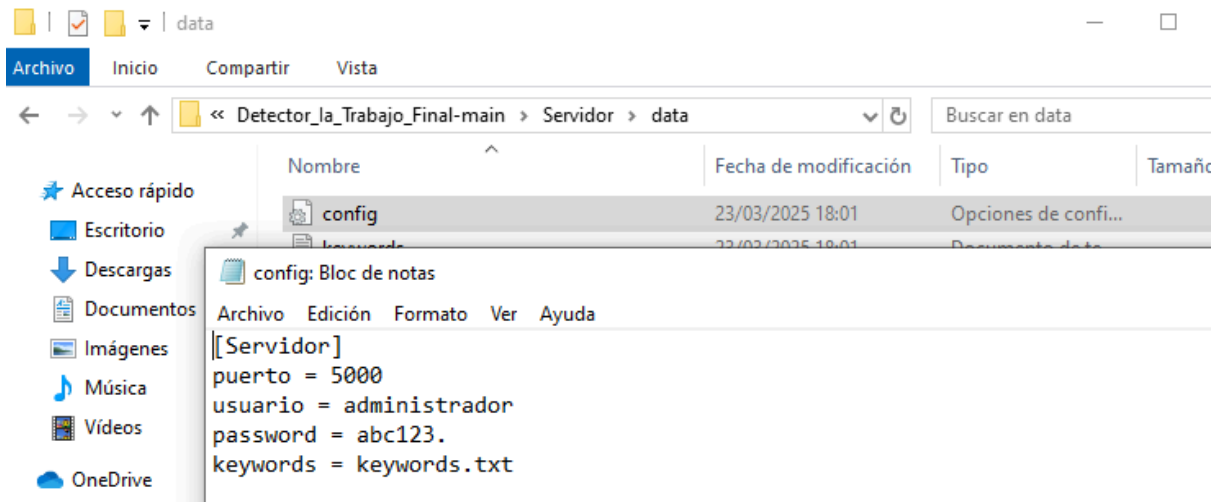
Anaconda PowerShell Prompt

```
(base) PS C:\Users\Tutorial>  
(base) PS C:\Users\Tutorial> conda activate detector_env  
(detector_env) PS C:\Users\Tutorial> _
```

- Nos dirigimos a la carpeta donde esta el servidor que descargamos de github anteriormente y ejecutamos `pip install -r requirements.txt` para instalar todo lo necesario

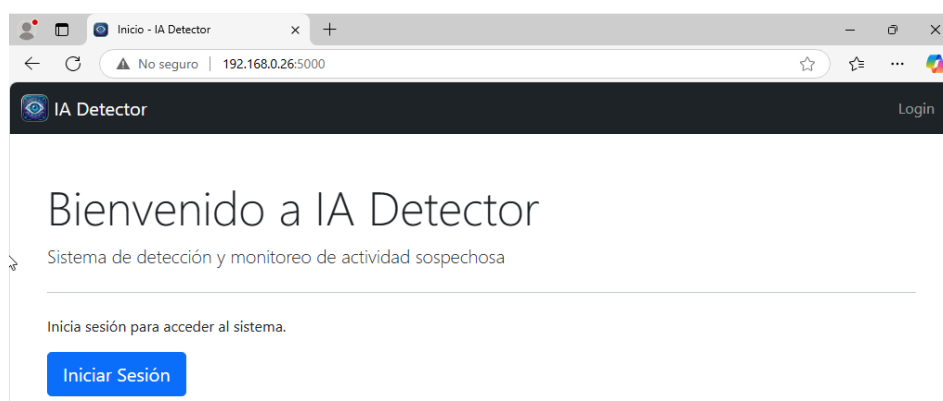
```
Anaconda PowerShell Prompt
(base) PS C:\Users\Tutorial> conda activate detector_env
(detector_env) PS C:\Users\Tutorial> cd C:\Users\Tutorial\Downloads\Detector_Ia_Trabajo_Final-main\Servidor
(detector_env) PS C:\Users\Tutorial\Downloads\Detector_Ia_Trabajo_Final-main\Servidor> pip install -r .\requirements.txt
```

- En la carpeta data del servidor modificaremos el archivo config.ini para cambiar la contraseña y el usuario del administrador y el puerto por donde se conectara nuestro cliente y nosotros.



- Ahora ejecutando el comando `python run.py` ejecutaremos el programa y ya estaría funcionando en tu localhost o en tu red local con tu ip y el puerto 5000 en nuestro caso 192.168.0.26:5000 ya podríamos acceder al panel de control y conectar nuestros clientes.

```
Selecionar Anaconda PowerShell Prompt
t-3.1.1 idna-3.10 importlib-metadata-8.5.0 itsdangerous-2.2.0 requests-2.32.3 typing-extensions-4.12.2 urllib3-2.2.3 z
3.20.2
(detector_env) PS C:\Users\Tutorial\Downloads\Detector_Ia_Trabajo_Final-main\Servidor> python run.py
Hilo de comprobación de horarios iniciado
* Serving Flask app 'run'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.0.26:5000
Press CTRL+C to quit
Comprobando horarios: Día 6 (Lunes=0), Hora 18:09
No se encontró horario activo para el momento actual
Próxima comprobación en 48.18 horas
```



2.1.2.Opción 2: Docker (recomendable).

Descargamos Docker Desktop <https://www.docker.com/products/docker-desktop/>

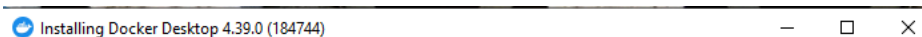
- Ejecutamos el instalador y seguimos los pasos instalara todo lo que necesite.



Unpacking files...

```
Unpacking file: frontend/locales/ms.pak
Unpacking file: frontend/locales/mr.pak
Unpacking file: frontend/locales/ml.pak
Unpacking file: frontend/locales/lv.pak
Unpacking file: frontend/locales/lt.pak
Unpacking file: frontend/locales/ko.pak
Unpacking file: frontend/locales/kn.pak
Unpacking file: frontend/locales/ja.pak
Unpacking file: frontend/locales/it.pak
Unpacking file: frontend/locales/id.pak
Unpacking file: frontend/locales/hu.pak
Unpacking file: frontend/locales/hr.pak
Unpacking file: frontend/locales/hi.pak
Unpacking file: frontend/locales/he.pak
```

- Al terminar aparecerá un mensaje para reiniciar el equipo, le damos y ya tendríamos docker instalado



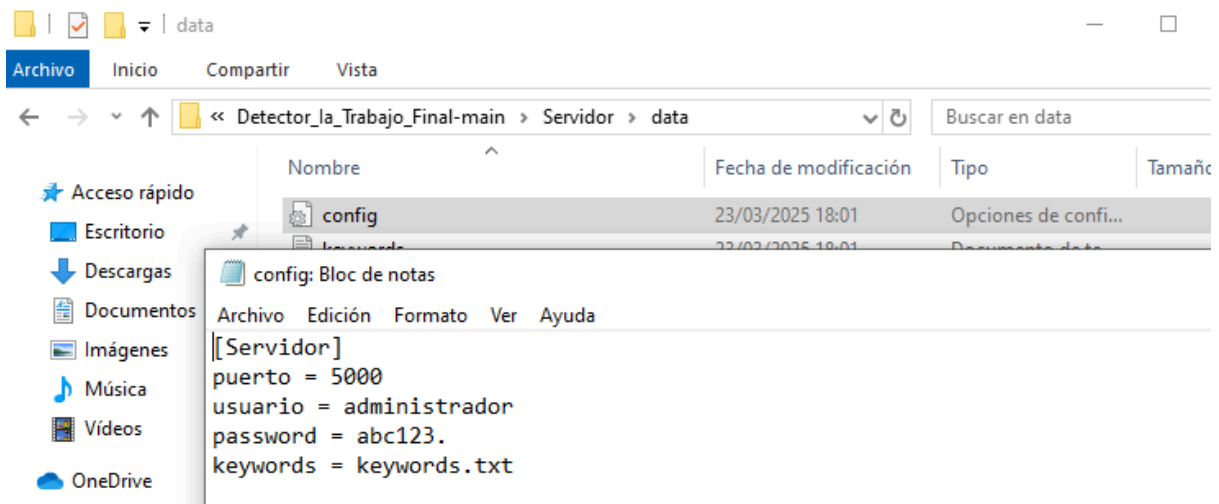
Docker Desktop 4.39.0

Installation succeeded

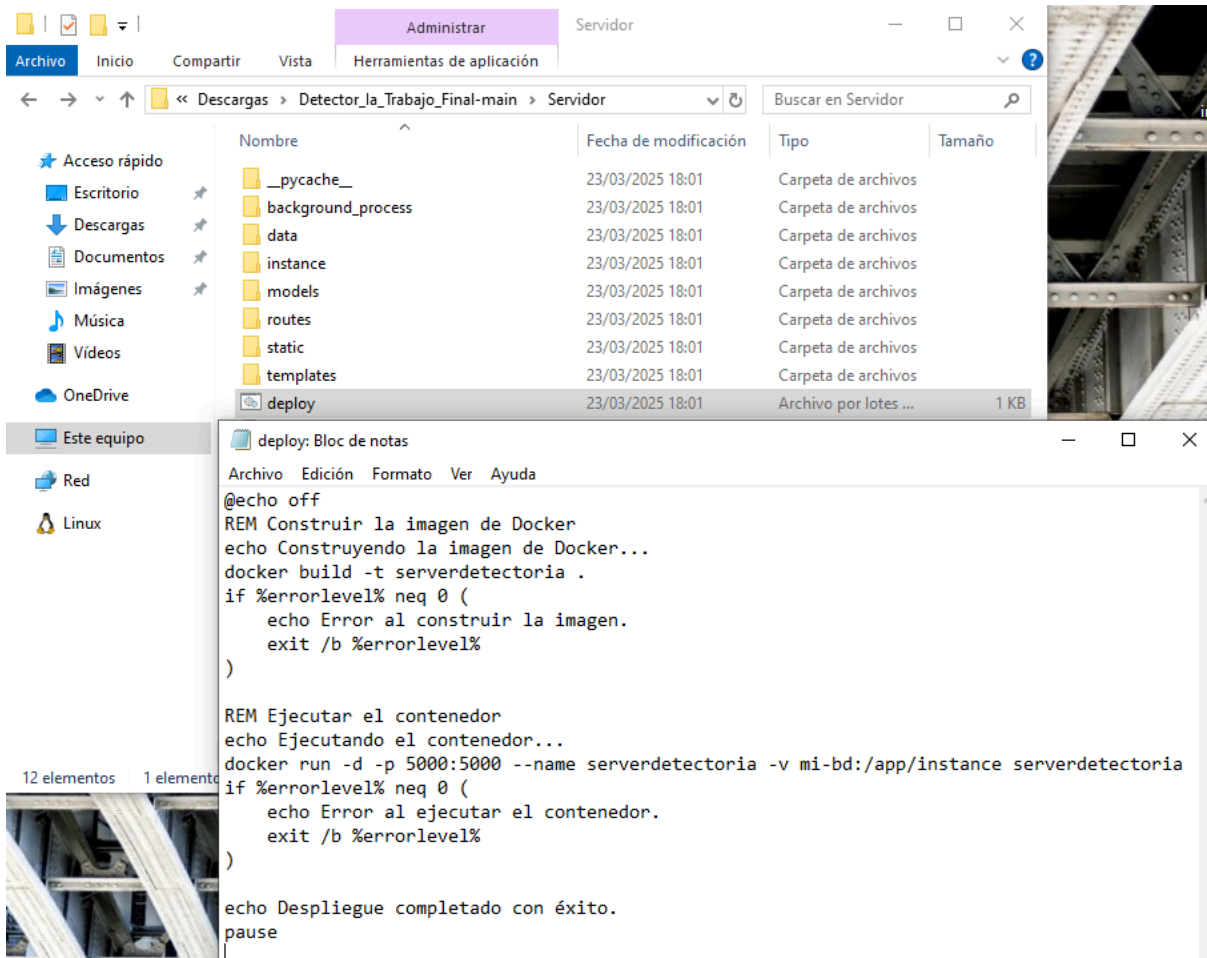
You must restart Windows to complete installation.

Close and restart

- Abrimos docker y nos dirigimos a la carpeta del servidor que descargamos de gitHub, en ella modificaremos en config.ini que se encuentra en la carpeta data. El puesto que pongamos es indiferente, yo recomiendo dejar el que está así la configuración será menor y más simple



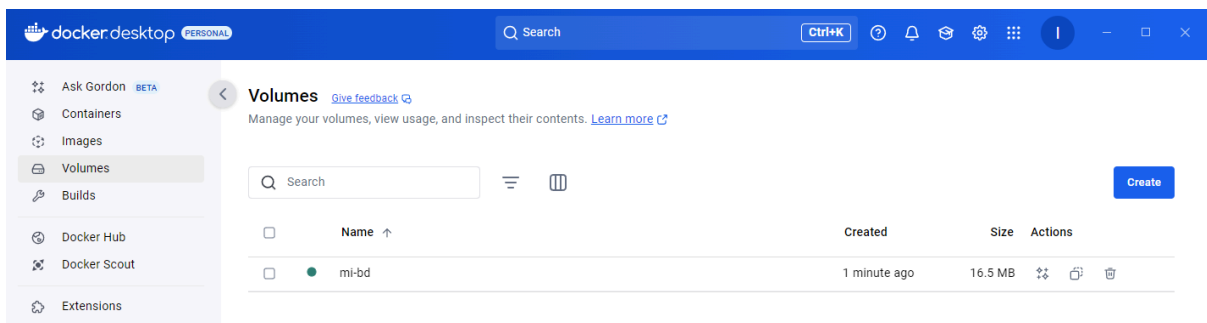
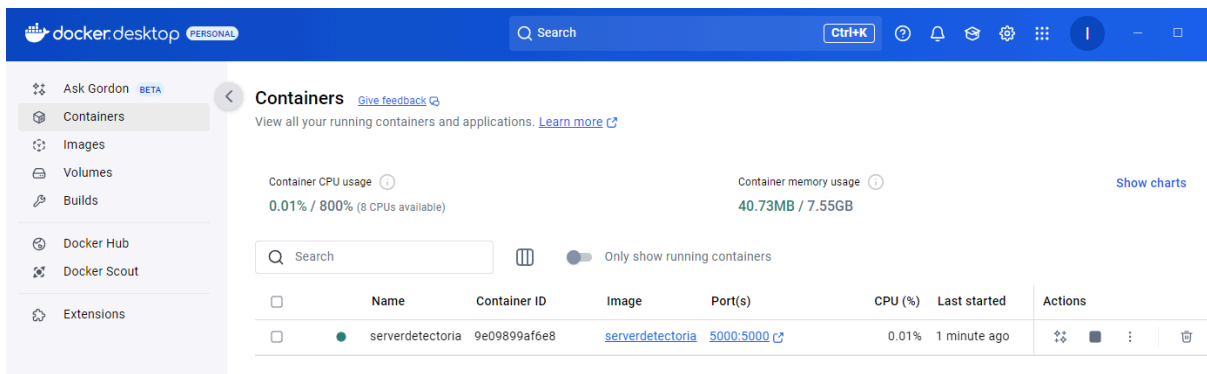
- Ahora vamos a editar el deploy.bat para configurar el puerto que queremos usar finalmente ya que el del config.ini será el interno del contenedor. El primer 5000 será nuestro puerto de enlace con el contenedor el otro es el interno ai colocamos el puerto que nos dé la gana y ejecutamos el script.



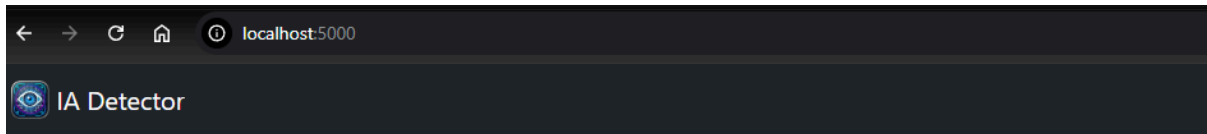
- Se creará nuestro contenedor automáticamente y un volumen para el archivo de la base de datos al mismo tiempo que se iniciara.

```
C:\Windows\system32\cmd.exe
Construyendo la imagen de Docker...
[+] Building 3.0s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 652B
=> [internal] load metadata for docker.io/library/python:3.8.20-slim
=> [auth] library/python:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/5] FROM docker.io/library/python:3.8.20-slim@sha256:1d52838af602b4b5a831beb13a0e4d073280665ea7be7f69ce2382
=> => resolve docker.io/library/python:3.8.20-slim@sha256:1d52838af602b4b5a831beb13a0e4d073280665ea7be7f69ce2382
=> [internal] load build context
=> => transferring context: 7.02kB
=> CACHED [2/5] WORKDIR /app
=> CACHED [3/5] COPY requirements.txt .
=> CACHED [4/5] RUN pip install --no-cache-dir -r requirements.txt
=> CACHED [5/5] COPY . .
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:8751c7255d513d036d5dbbc15b9753704de520865ad1abde52b57aa26866e2df
=> => exporting config sha256:f071b38d977d4877dc39ed5fafd8f91fbb21b43d7baa2bf323a3cd057c08359b
=> => exporting attestation manifest sha256:472ea297ada33811777ab107427abc06d9cb4a7576da16441695ae5d3b9e666c
=> => exporting manifest list sha256:8ddd48c8eb15c06108a31e5e5b98b9ae5548ac051b41298db619e95c49c9ac45
=> => naming to docker.io/library/serverdetectoria:latest
=> => unpacking to docker.io/library/serverdetectoria:latest

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/vgrmx4abrzyu82ptduf9p2m01
Ejecutando el contenedor...
9e09899af6e8df19f4716e49a30033f15ca372970273d056519683f1dceb2e6f
Despliegue completado con éxito.
Presione una tecla para continuar . . .
```



- Poniendo tu ip o localhost + el puerto en este caso 5000 podremos acceder al software.



Bienvenido a IA Detector

Sistema de detección y monitoreo de actividad sospechosa

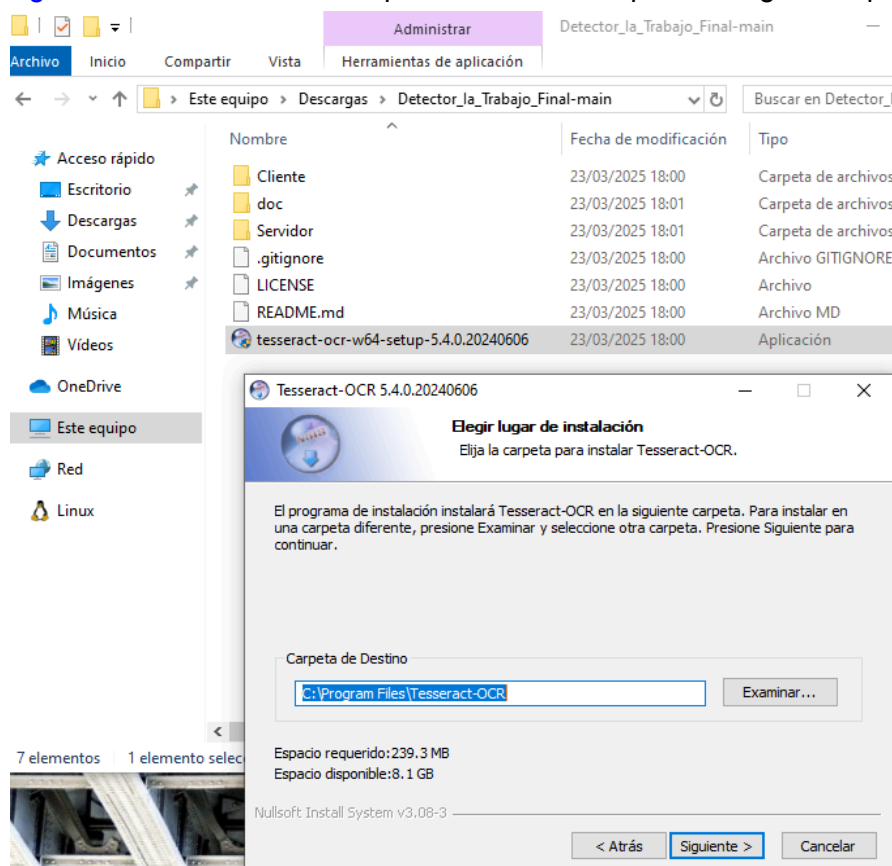
Inicia sesión para acceder al sistema.

Iniciar Sesión









2.2.Cliente

2.2.1.Anaconda (entorno virtual).

- En la carpeta del proyecto que descargamos de gitHub ejecutamos el tesseract, necesario para convertir las imágenes a texto. Seguimos los pasos del instalador y nos quedaremos con la ruta donde se ha instalado en este caso es <C:\Program Files\Tesseract-OCR> que necesitaremos para configurar el programa



- En la carpeta de Cliente editaremos el config.ini y pegaremos la ruta del tesseract justo con la ip y el puerto de nuestro servidor

Nombre	Fecha de modificación	Tipo	Tamaño
 build	23/03/2025 18:00	Carpeta de archivos	
 dist	23/03/2025 18:01	Carpeta de archivos	
 cliente.py	23/03/2025 18:00	Archivo PY	9 KB
 cliente_debug.spec	23/03/2025 18:00	Archivo SPEC	1 KB
 cliente_hidden.spec	23/03/2025 18:00	Archivo SPEC	1 KB
 config	23/03/2025 18:00	Opciones de confi...	1 KB
 keywords.pkl	23/03/2025 18:00	Archivo PKL	1 KB
 requirements	23/03/2025 18:00	Documento de te...	1 KB

 config: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
[Cliente]
url_servidor = http://192.168.0.14:5000
ruta = C:\\Program Files\\Tesseract-OCR\\tesseract.exe
```

- Para la instalación de anaconda seguir el punto **2.1.1.Opción 1: Anaconda (entorno virtual)**. del servidor en el apartado anterior.
- Abrimos una consola de Anaconda y con `cd` nos situamos en la carpeta del cliente y activamos el entorno de conda que creamos

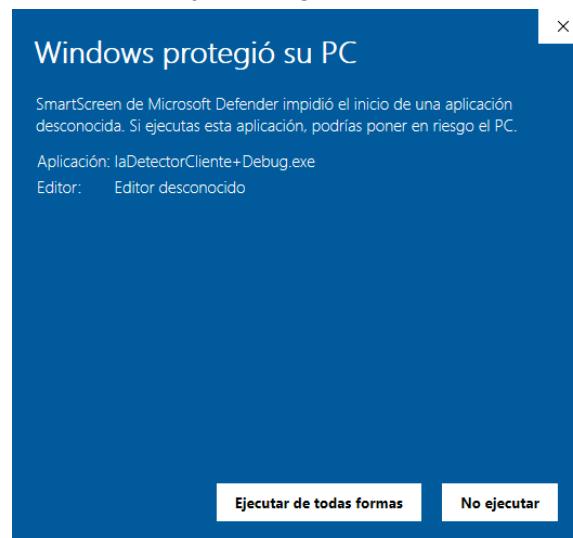
```
Anaconda PowerShell Prompt
(base) PS C:\Users\Tutorial> cd C:\Users\Tutorial\Downloads\Detector_Ia_Trabajo_Final-main\Cliente
(base) PS C:\Users\Tutorial\Downloads\Detector_Ia_Trabajo_Final-main\Cliente> conda activate detector_env
(detector_env) PS C:\Users\Tutorial\Downloads\Detector_Ia_Trabajo_Final-main\Cliente> _
```

- Ejecutamos `pip instal -r requirements.txt` para instalar todas las librerías necesarias
- Ejecutamos el comando `python cliente.py` y el programa ya se pondría a correr

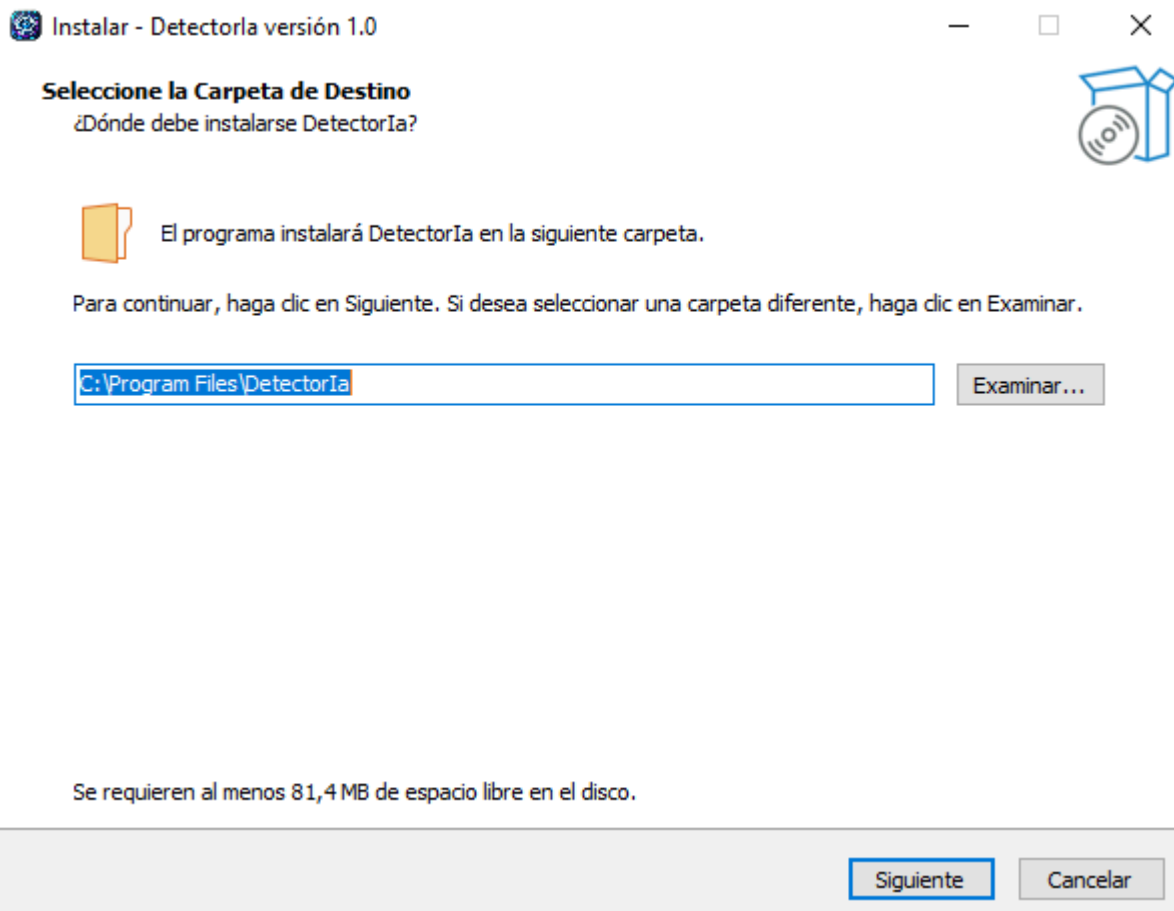
```
Anaconda PowerShell Prompt
Successfully built PyAutoGUI PyGetWindow pycreeze pytweneing mouseinfo pymsgbox pyrect pyperclip
Installing collected packages: pytweneing, pyrect, pyperclip, pymsgbox, pycreeze, PyGetWindow, packaging, mouseinfo, py
tesseract, PyAutoGUI
Successfully installed PyAutoGUI-0.9.54 PyGetWindow-0.0.9 mouseinfo-0.1.3 packaging-24.2 pymsgbox-1.0.9 pyperclip-1.9.0
pyrect-0.2.0 pycreeze-1.0.1 pytesseract-0.3.13 pytweneing-1.2.0
(detector_env) PS C:\Users\Tutorial\Downloads\Detector_Ia_Trabajo_Final-main\Cliente> python .\cliente.py
[2025-03-23 19:21:20.206011] Intentando descargar las palabras clave desde: http://192.168.0.18:5000/keywords
[2025-03-23 19:21:20.251495] Archivo de palabras clave descargado exitosamente.
[2025-03-23 19:21:20.251495] Cambio de ventana detectado: Anaconda PowerShell Prompt
['chatgpt', 'copilot', 'openai', 'bard', 'codeium', 'qwen', 'deepseek', 'davinci', 'codex', 'watson', 'lumen5', 'hivemin
d', 'chatbot', 'deepmind', 'cortex', 'automated', 'ai-based', 'ml-powered', 'ai-driven']
```

2.2.2.Instalador (recomendable).

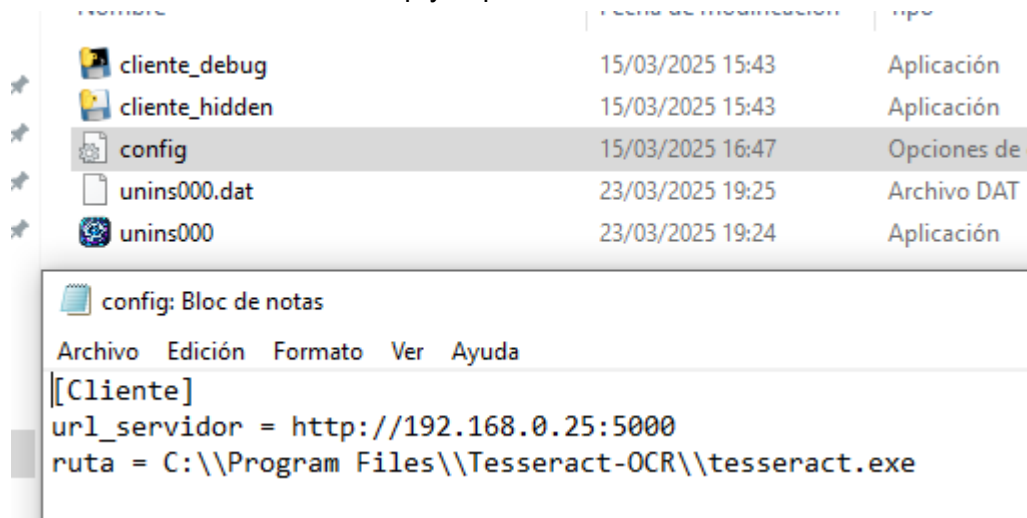
- Ejecutamos el instalador laDetectorCliente+Debug.exe y ya nos instalará el tesseract automáticamente, además nos copiara un ejecutable del cliente. Nos va a decir que es peligroso pero le damos a ejecutar igualmente.



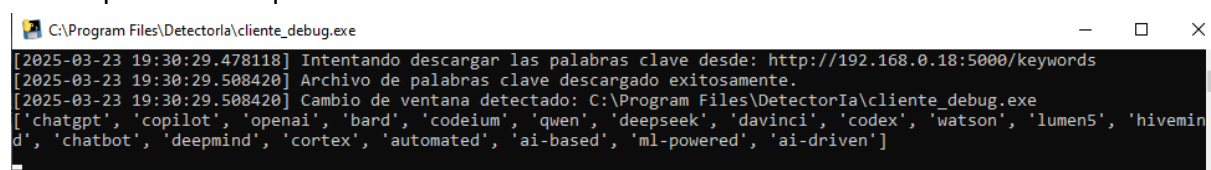
- Elegimos donde queremos copiar nuestro programa. Deberemos ir ai mas tarde para configurar el config.ini.



- nos dirigimos a la ruta de instalación y configuraremos el config.ini con la ruta del tesseract además de la ip y el puerto del servidor



- para ejecutar el cliente tenemos 2 opciones el _hidden que lo ejecutara en segundo plano sin consola o el _debug que tendremos una consola para ver qué problemas pueden estar pasando ejecutaremos como administrador el debug para probar y ver por consola que se conecta correctamente.



3.Despliegue del sistema

Configuración del entorno y ejecución en producción.

3.1.Servidor

El despliegue del servidor se realiza utilizando Docker, lo que permite ejecutar la aplicación de forma aislada, controlada y automática.

Gracias al script de despliegue incluido en el proyecto, el proceso de puesta en marcha es totalmente automático y solo requiere una configuración inicial mínima.

¿Por qué se usa Docker?

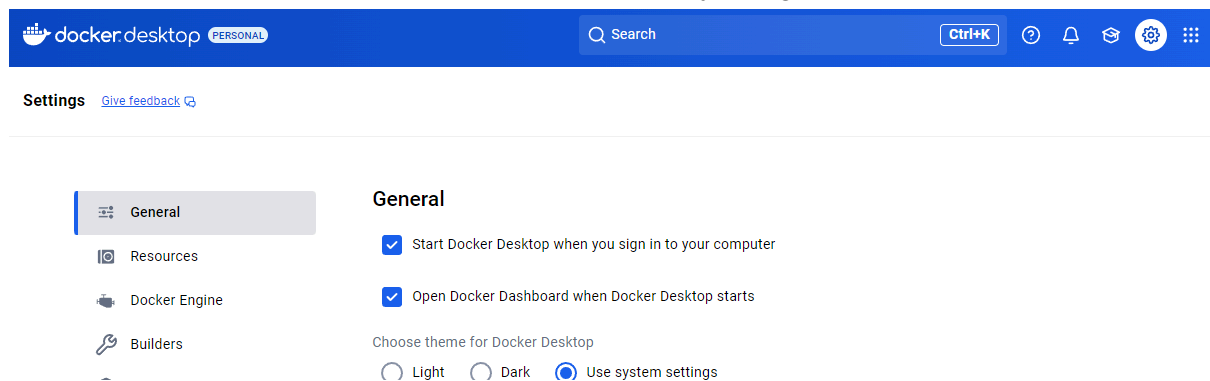
- Docker permite:
 - Ejecutar la aplicación con todas sus dependencias sin preocuparse por el sistema operativo.
 - Asegurar que el entorno de ejecución sea siempre el mismo (consistencia entre máquinas).
 - Controlar de forma sencilla el estado del servidor, incluso después de reinicios del sistema.

¿Qué hace exactamente el despliegue?

- Al ejecutar el script desplegar servidor.bat, se realiza lo siguiente:
 - Se construye la imagen Docker del servidor.
 - Se elimina cualquier contenedor anterior para evitar errores.
 - Se lanza el contenedor en segundo plano, con una configuración especial:
 - Se expone el puerto 5000 (puerto de Flask).
 - Se usa un volumen persistente para mantener los datos (base de datos SQLite).
 - Se aplica la política --restart=always, lo que significa que Docker reiniciará automáticamente el contenedor si se apaga o reinicia el sistema.

¿Qué se necesita hacer para que se inicie automáticamente?

- Nada más que asegurarse de que Docker Desktop está configurado para arrancar con Windows.
- Esto se hace una sola vez:
 - Abrir Docker Desktop.
 - Ir a Settings > General.
 - Activar la opción "Start Docker Desktop when you log in".



Con esto, no es necesario programar tareas adicionales, ni accesos directos, ni scripts extra.

Al encender el ordenador, Docker arranca, y con él se lanza automáticamente el servidor dentro del contenedor.

3.2. Cliente

El despliegue del cliente se basa en la instalación de un ejecutable personalizado que se ejecuta de forma automática cada vez que se inicia el sistema. Este ejecutable, llamado [cliente_hidden.exe](#), se copia en la ruta [C:\Program Files\Cliente\cliente_hidden.exe](#) mediante el instalador proporcionado.

Una vez instalado, es necesario crear una tarea programada en Windows que permite ejecutar el cliente en segundo plano de manera automática con cada arranque del sistema, sin necesidad de intervención por parte del usuario.

Ejecución automática al iniciar Windows

Para lograr que el cliente se inicie al encender el equipo y se ejecute con permisos elevados, se configura una tarea programada utilizando la cuenta del sistema (SYSTEM).

Esto permite que la aplicación se ejecute con privilegios de administrador sin mostrar mensajes de confirmación ni requerir credenciales.

La tarea se crea ejecutando el siguiente comando en PowerShell, con permisos de administrador:

```
schtasks /create /tn "ClientelA" /tr "\"C:\Program Files\Cliente\cliente_hidden.exe\""" /sc  
onstart /ru "SYSTEM" /rl HIGHEST /f
```

Este comando registra una nueva tarea programada llamada ClientelA que:

- Se ejecuta automáticamente al iniciar el sistema.
- Utiliza la cuenta interna SYSTEM, por lo que no requiere contraseña ni intervención del usuario.
- Se ejecuta con el nivel más alto de privilegios disponibles.

Con esta configuración, el cliente queda desplegado como un proceso oculto y persistente que se activa de forma inmediata al arrancar el sistema operativo.

¿Cómo eliminar la tarea si hiciera falta?

Ejecuta este comando.

```
schtasks /delete /tn "ClientelA" /f
```

Conclusión

El cliente se instala mediante un ejecutable personalizado que coloca el archivo necesario en una ruta protegida del sistema. Una vez instalado, y tras ejecutar una única vez el comando indicado en PowerShell con permisos de administrador, el cliente queda configurado para iniciarse de forma automática en cada arranque de Windows, con permisos elevados y sin necesidad de intervención del usuario. Esto asegura su correcto funcionamiento como herramienta de monitorización en segundo plano.

Manual de usuario

Este manual explica cómo utilizar el sistema para la detección de inteligencia artificial en exámenes y clases. Está dirigido a administradores y profesores, quienes gestionan y supervisan el monitoreo de los equipos en el aula.

1. Arquitectura general del sistema

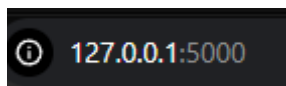
El sistema está compuesto por tres elementos clave: el servidor, la interfaz web y el cliente. Cada uno cumple una función específica y es necesario que al menos el servidor esté activo para que la aplicación funcione correctamente.

- **Servidor:** Es el núcleo del sistema. Gestiona la base de datos, expone las API y mantiene activa la lógica de negocio. Es indispensable que esté en ejecución para que tanto la interfaz web como el cliente pueda comunicarse con él.
- **Interfaz web:** Permite a los usuarios (profesores o administradores) acceder al sistema para gestionar usuarios, visualizar capturas, configurar horarios, etc. La interfaz web depende completamente del servidor. Si este no está activo, la interfaz no podrá cargar ni acceder a los datos.
- **Cliente:** Es una aplicación que debe estar en ejecución en los ordenadores que se desean monitorizar. Su función es tomar capturas de pantalla, analizarlas y, si detecta actividad sospechosa, enviarlas automáticamente al servidor. Aunque la interfaz web puede funcionar sin el cliente, no se recibirán nuevas capturas si el cliente no está activo.

Por tanto, para el funcionamiento completo del sistema (captura, análisis y visualización), se requiere que el servidor y el cliente estén en ejecución. La interfaz web actúa como herramienta de visualización y control, pero depende totalmente del servidor para operar.

2. Acceso y autenticación


Abrir un navegador web e ingresar la dirección del servidor proporcionada por el administrador.



Introducir el usuario y contraseña en la pantalla de inicio de sesión.

IA Detector

Login



Inicio de sesión

Usuario

Usuario

Contraseña

Contraseña

Iniciar sesión

© 2025

Según el rol del usuario, se mostrará el panel correspondiente:

- Administrador: Acceso completo a la gestión del sistema.
- Profesor: Acceso a la visualización de capturas y gestión de palabras clav

3.Funciones según el rol del usuario

3.1.Para administradores

3.1.1.Gestión de usuarios:

Crear, modificar y eliminar cuentas de profesores.

- Observar todos los usuarios.

IA Detector

Panel de Control Usuarios Editar Keywords Horarios administrador Cerrar Sesión

Lista de Usuarios

Nuevo Usuario

ID	Nombre	Administrador	Acciones	
1	administrador	Si		
2	Janeiro	No	Cambiar Contraseña	Eliminar
3	ivanB	No	Cambiar Contraseña	Eliminar
4	Javier	No	Cambiar Contraseña	Eliminar
5	Julia	No	Cambiar Contraseña	Eliminar

- En el botón de Eliminar borraremos el usuario y todas sus capturas de pantalla

Eliminar

- El botón de cambiar contraseña nos abrirá un formulario para introducir una nueva contraseña aunque no sepas la anterior

Cambiar Contraseña

The screenshot shows the 'Lista de Usuarios' (List of Users) page in the 'IA Detector' application. A modal titled 'Cambiar Contraseña' (Change Password) is open, allowing the user to change the password for the user 'ivanB'. The modal contains two input fields: 'Nueva Contraseña' (New Password) and 'Confirmar Nueva Contraseña' (Confirm New Password). Below the fields are 'Cancelar' (Cancel) and 'Guardar Cambios' (Save Changes) buttons. In the background, a table lists users with columns 'ID' and 'Nombre'.

ID	Nombre
1	administrador
2	Janeiro
3	ivanB
4	Javier
5	Julia

- Nuevo usuario nos llevará a una nueva ventana donde tendremos que introducir los datos del nuevo usuario

The screenshot shows the 'Registro de Usuario' (User Registration) form. It includes a header with the application name 'IA Detector' and navigation links. The form has three input fields: 'Nombre de usuario' (Username) with a placeholder 'Usuario', 'Contraseña' (Password) with a placeholder 'Contraseña', and 'Confirmar contraseña' (Confirm password) with a placeholder 'Confirmar contraseña'. A blue 'Registrarse' (Register) button is at the bottom. The footer shows '© 2025'.

3.1.2. Configuración de horarios:

Establecer los periodos en los que el sistema realiza capturas automáticas.

Gestión de Horarios

Añadir Nuevo Horario

Usuario

Día

Hora inicio

Hora fin

Seleccionar usuario

Lunes

--:--

--:--

Añadir Horario

Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
Usuario	Hora inicio	Hora fin	Acciones			
Janeiro	18:20	18:22	Eliminar			
ivanB	18:22	18:25	Eliminar			
Javier	18:25	18:27	Eliminar			
ivanB	18:37	18:40	Eliminar			
Janeiro	18:45	18:47	Eliminar			
ivanB	18:47	18:49	Eliminar			
Javier	18:50	18:53	Eliminar			

- Seleccionaremos al usuario en el desplegable.

Gestión de Horarios

Añadir Nuevo Horario

Usuario

Día

Seleccionar usuario

Seleccionar usuario

Janeiro

ivanB

Javier

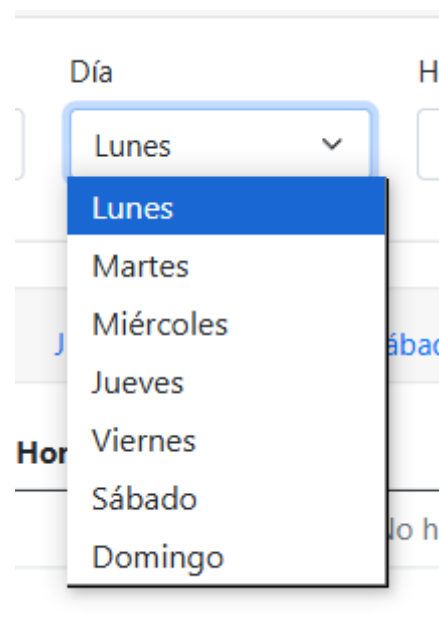
Julia

Lunes

Jueves

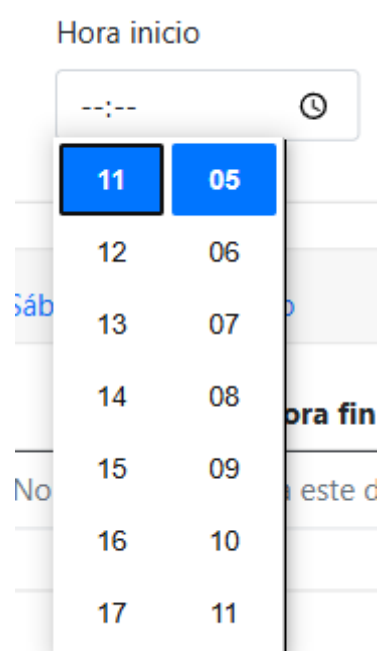
ora inici

- Seleccionaremos el día de la semana.



A screenshot of a web form showing a dropdown menu for selecting a day of the week. The label "Día" is above the dropdown. The dropdown is open, showing a list of days: Lunes, Martes, Miércoles, Jueves, Viernes, Sábado, and Domingo. "Lunes" is selected and highlighted in blue. The background shows parts of other form fields like "Hora inicio" and "Hora fin".

- Seleccionamos hora de inicio



A screenshot of a web form showing a time selection interface. The label "Hora inicio" is above the interface. The interface consists of a dropdown menu showing "--:--" and a clock icon. Below the dropdown is a table with two columns for hours and minutes. The first row, "11" and "05", is highlighted in blue. The background shows parts of other form fields like "Sáb" and "hora fin".

Hour	Minute
11	05
12	06
13	07
14	08
15	09
16	10
17	11

- Seleccionamos hora de fin

Hora fin

--:--

🕒

11	06
12	07
13	08
14	09
15	10
16	11
17	12

- Pulsamos el botón de añadir horario. OJO nunca podrá haber 2 profesores en el mismo horario, te salta un error y el horario no se creará.

Añadir Horario


- En la tabla de abajo podremos seleccionar el día para ver los horarios que están creados y para quién están creados además de eliminarlos.

<div> Lunes Martes Miércoles Jueves Viernes Sábado Domingo </div>			
Usuario	Hora inicio	Hora fin	Acciones
Janeiro	18:20	18:22	Eliminar
ivanB	18:22	18:25	Eliminar
Javier	18:25	18:27	Eliminar
ivanB	18:37	18:40	Eliminar
Janeiro	18:45	18:47	Eliminar
ivanB	18:47	18:49	Eliminar
Javier	18:50	18:53	Eliminar

3.1.3. Visualización de capturas:

Consultar y gestionar las capturas realizadas.

- Entrar en el Panel de Control, ahí visualizamos todos los equipos que alguna vez han registrado una imagen.

 IA Detector

Panel de Control | Usuarios | Editar Keywords | Horarios | administrador | Cerrar Sesión


Panel de Control

Control de Capturas


☒ Modo Automático

Estado del Sistema: Desactivado

Capturas por Equipo



I7-4790K
5 días



PORATILPOCHO
1 días

- Control de Capturas podemos cambiar de automático a manual y además ver para quien se están guardando las capturas en ese momento.
 - Modo Automático

Control de Capturas

☒ Modo Automático

Estado del Sistema: Desactivado

Control de Capturas

☒ Modo Automático

Estado del Sistema: Activado
Guardando para: **ivanB** (Modo Automático)

- Modo manual

Control de Capturas

☐ Modo Automático

Estado del Sistema: Activado
Guardando para: **administrador** (Modo Manual)

Detener Capturas



Control de Capturas

☐ Modo Automático

Estado del Sistema: Desactivado

Iniciar Capturas


- Capturas por Equipo podremos clicar en un equipo cualquiera, también nos da la información de cuántos días tiene capturas guardadas.

Capturas por Equipo	
	
I7-4790K	PORATILPOCHO
5 días	1 días

- En el equipo podemos ver todos los días separados por carpetas, entrar en uno y ver las capturas.


[Panel de Control](#) / I7-4790K

Capturas de I7-4790K




2025-03-21
7 capturas

Eliminar carpeta




2025-03-18
1 capturas

Eliminar carpeta




2025-03-16
6 capturas

Eliminar carpeta



2025-03-13
8 capturas

Eliminar carpeta




2025-03-12
9 capturas

Eliminar carpeta

[Panel de Control](#) / [I7-4790K](#) / 2025-03-12

Capturas del 2025-03-12


20:39:35



Ordenador: I7-4790K Ventana: cliente.py - cliente - Visual Studio Code Texto detectado: %J File Edit...

Imagen Texto Eliminar


20:39:21



Ordenador: I7-4790K Ventana: cliente.py - cliente - Visual Studio Code Texto detectado: %J File Edit...

Imagen Texto Eliminar


20:38:40



Ordenador: I7-4790K Ventana: cliente.py - cliente - Visual Studio Code Texto detectado: %3 File Edit...

Imagen Texto Eliminar


20:38:29



Ordenador: I7-4790K Ventana: cliente.py - cliente - Visual Studio Code Texto detectado: %J File Edit...

Imagen Texto Eliminar


20:30:12



Fecha: 20250312_203012 Cliente: I7-4790K Ventana: cliente.py - cliente - Visual Studio Code Texto de...

Imagen Texto Eliminar

20:29:57




Fecha: 20250312_202957 Cliente: I7-4790K Ventana: config.ini - cliente - Visual Studio Code Texto de...

Imagen Texto Eliminar

- En el botón de imagen podremos descargar la Imagen, en el de Texto el .txt y en el botón Eliminar borraremos la captura de la base de datos.

3.1.4. Editar los keywords:

Consultar y modificar las palabras clave utilizadas para la detección de IA.

 IA Detector

Panel de Control Usuarios Editar Keywords Horarios administrador Cerrar Sesión

Editar Palabras Clave

Descargar Archivo

Contenido del archivo keywords.txt:

chatgpt
copilot
openai
bard
codeium
qwen
deepseek
davinci
codex
watson
lumen5
hivemind
chatbot
deepmind
cortex
automated
ai-based
ml-powered
ai-driven

Guardar Cambios

- En el recuadro podremos escribir más palabras clave y darle a guardar, recomendable escribir una palabra clave por cada línea
- En el botón de Descargar Archivo descargamos el .txt por si lo queremos sustituir en otro equipo.

3.2. Para profesores

La parte de panel de control a la única que tiene acceso un profesor, el uso es igual que en el administrador.

- Visualización de capturas: Acceder a las imágenes capturadas por los clientes asignados.
- Descarga y eliminación de capturas: Gestionar las imágenes almacenadas en el sistema.
- Modo manual de capturas: Habilitar o deshabilitar la captura automática dentro de su sesión.
- Descarga y eliminación de capturas: Gestionar las imágenes almacenadas en el sistema.

4. Cierre de sesión

Al cerrar sesión te llevará de vuelta al login, el programa seguirá funcionando y modificando los horarios con normalidad.

Bibliografía

Fuentes utilizadas en formato APA 7.^a edición.

Referencias

- Anaconda Inc. (s.f.). Anaconda Documentation. Recuperado de <https://docs.anaconda.com/>
- Bootstrap. (s.f.). Bootstrap Documentation. Recuperado de <https://getbootstrap.com/>
- Codigofacilito. (s.f.). Cursos de programación y desarrollo web. Recuperado de <https://codigofacilito.com/>
- Dev.to. (s.f.). Comunidad de desarrolladores. Recuperado de <https://dev.to/>
- Docker Inc. (s.f.). Docker Documentation. Recuperado de <https://docs.docker.com/>
- Google. (s.f.). Tesseract OCR Documentation. Recuperado de <https://github.com/tesseract-ocr/tesseract>
- Medium. (s.f.). Artículos de tecnología y desarrollo. Recuperado de <https://medium.com/>
- OpenWebinars. (s.f.). Formación en tecnologías de la información. Recuperado de <https://openwebinars.net/>
- Pallets. (s.f.). Flask Documentation. Recuperado de <https://flask.palletsprojects.com/>
- SQLAlchemy. (s.f.). SQLAlchemy Documentation. Recuperado de <https://docs.sqlalchemy.org/>

Stack Overflow. (s.f.). Foro de desarrollo y programación. Recuperado de
<https://stackoverflow.com/>

Sweigart, A. (s.f.). PyAutoGUI Documentation. Recuperado de
<https://pyautogui.readthedocs.io/>

Esta bibliografía ha sido fundamental para la correcta implementación del proyecto, permitiendo solucionar problemas técnicos y mejorar la calidad del desarrollo.