

REPORTE DE INVESTIGACIÓN Y PROPUESTAS DE MEJORA

Nombre: Andres Reyes Sanchez // Código: TG24

Área: División de Programación de Cobranza

1. INTRODUCCIÓN

El propósito es identificar soluciones que optimicen las tareas repetitivas y mejoren la eficiencia operativa, complementando las funciones actuales con procesos automáticos que puedan ejecutarse tanto en primer plano, como en segundo plano (dependiendo de la operatividad y posibles soluciones) sin interrumpir otras actividades del sistema.

2. INVESTIGACIÓN DE LIBRERÍAS Y HERRAMIENTAS EN PYTHON

2.1. Automatización de Procesos

Se identificaron librerías útiles para automatizar tareas repetitivas tanto en escritorio como en entornos web:

- pywinauto: Automatiza la interacción con aplicaciones de escritorio (útil para procesos internos o sistemas como RSIRAT).
- selenium y playwright: Permiten automatizar la navegación en portales web, enviar formularios, realizar consultas y validar datos en masa.
- pyautogui: Facilita la simulación de movimientos del mouse y teclado; ideal para automatizaciones visuales o simples.

Además, estos scripts pueden complementarse con módulos como subprocess, threading o multiprocessing, permitiendo que se ejecuten en segundo plano mientras el usuario realiza otras tareas.

Esto es especialmente útil en entornos con múltiples procesos (por ejemplo, automatizar consultas de datos mientras se gestionan reportes).

Esto permite mantener la interfaz libre para otras tareas y facilita la integración con sistemas o bases de datos sin bloquear recursos.

2.2. Extracción y Procesamiento de Datos Web (Web Scraping)

Para recolectar grandes volúmenes de información desde portales públicos o internos:

- requests + BeautifulSoup: Eficientes para obtener y analizar datos HTML.
- Scrapy: Framework escalable para scraping masivo, con soporte para múltiples hilos y ejecución continua en segundo plano.
- Pandas + Dask: Procesamiento y análisis de datos con soporte para volúmenes grandes, distribuyendo la carga en varios núcleos.
- SQLAlchemy: Facilita la conexión directa con bases de datos SQL, permitiendo registrar los datos extraídos de forma estructurada.

Estos procesos pueden ejecutarse de manera programada (por ejemplo, con Windows Task Scheduler o cron jobs) para mantener actualizados los datos automáticamente, sin requerir intervención humana.

3. Análisis de Propuesta: Implementación de IA en Llamadas de Cobranza

3.1. Objetivo

Explorar el uso de inteligencia artificial (IA) para:

- Automatizar llamadas de cobranza.
- Registrar y transcribir las conversaciones.
- Analizar emociones, palabras clave y resultados.
- Permitir que las llamadas sean procesadas y registradas en segundo plano, evitando cargas manuales.

3.2. Tecnologías Recomendadas

Tecnología	Descripción	Posible Aplicación
Speech-to-Text (STT)	Convierte voz a texto. Ejemplos: <i>Google Speech API</i> , <i>Whisper (OpenAI)</i> , <i>Amazon Transcribe</i> .	Transcripción automática de llamadas.
Chatbots de Voz (TTS + IA Conversacional)	Emplea modelos como GPT o Dialogflow para mantener conversaciones naturales.	Llamadas automáticas con guiones de cobranza.
Análisis de Sentimiento (NLP)	Usa modelos NLP para detectar emociones y temas clave. Ejemplo: <i>Transformers (HuggingFace)</i> , <i>Amazon Comprehend</i> .	Evaluación del tono y efectividad de las llamadas.
Automatización de Procesos en Segundo Plano	Scripts que graban, transcriben y analizan sin intervención humana.	Ejecución continua y registro automático en base de datos.

3.3. Flujo Propuesto

- Inicio automático de llamadas mediante IA con script de cobranza.
- Grabación de audio entrante y saliente.
- Conversión a texto (STT).
- Análisis semántico y emocional de la conversación.
- Registro automático en base de datos SQL o en la nube.
- Ejecución continua en segundo plano, permitiendo que el sistema atienda múltiples llamadas simultáneamente.
- Visualización en Power BI o Streamlit con métricas de efectividad, duración y resultados.

4. Beneficios Esperados

- Reducción de la carga manual y repetitiva.
- Mayor trazabilidad y registro de las interacciones con contribuyentes.
- Posibilidad de ejecución en segundo plano para mantener otros procesos activos.
- Mejora en la eficiencia y control del área de cobranzas.
- Capacidad de análisis avanzado mediante IA (Inteligencia artificial) y NLP (Procesamiento del lenguaje natural).

5. Conclusiones

El uso de Python, junto con librerías de automatización, extracción de datos y modelos de inteligencia artificial, puede modernizar significativamente los procesos de cobranza en la SUNAT.

Además, el uso de ejecuciones en segundo plano permite mantener la operatividad continua de los sistemas, optimizando recursos y tiempo.

La integración de un sistema de transcripción, análisis y automatización de llamadas representa una innovación viable y alineada con la transformación digital institucional.