Documentación Técnica:Web Scraper para plataforma X

**Autores:** Leonardo Valdés Palafox, Andrea Marlene Ortega Almendares

**Estándar:** *Este documento sigue el estándar* ***IEEE 829-2008*** *para Documentación de Software*

**Versión: 2.0**

**Fecha: 30 de marzo de 2025**

# Tabla de Contenidos

1. Introducción

 Propósito  Alcance

1. Arquitectura del Sistema

 Diagrama de Arquitectura

1. Componentes Principales  Clase TwitterScraper

 Funciones Auxiliares

1. Flujo de Trabajo

 Extracción por cuenta

 Extracción de datos por publicación  Almacenamiento

1. Funcionalidades Clave

 Extracción de Estadísticas  Filtrado por Antigüedad

 Detección de Contenido Multimedia

1. Mecanismos Anti-Detección

 Configuración del navegador  Comportamiento humanizado  Manejo de popups

1. Manejo de Datos

 Estructura de Datos  Almacenamiento

1. Limitaciones Conocidas
2. Requisitos Técnicos

 Dependencias

 Entorno de Ejecución

10. Referencias

11. Apéndice: Diagramas UML  Diagrama de Clases

 Diagrama de Flujo

 Diagrama de Secuencia

 Diagrama de Componentes  Diagrama de Despliegue

# Introducción

Este documento proporciona una descripción técnica detallada del "Web Scraper para plataforma X", una herramienta diseñada para extraer datos de la plataforma X (anteriormente Twitter) de manera automatizada y eficiente, sin necesidad de utilizar la API oficial. El sistema está desarrollado en Python y utiliza Selenium para interactuar con la interfaz web.

## Propósito

El propósito de esta herramienta es permitir la recolección de datos públicos de cuentas específicas en la plataforma X, incluyendo texto de publicaciones, estadísticas de interacción (comentarios, retweets, me gusta, compartidos) y metadatos adicionales como fechas y URLs.

## Alcance

El sistema permite:

 Extraer datos de múltiples cuentas en una sola ejecución  Filtrar publicaciones por antigüedad (menos de dos años)  Detectar contenido multimedia en las publicaciones

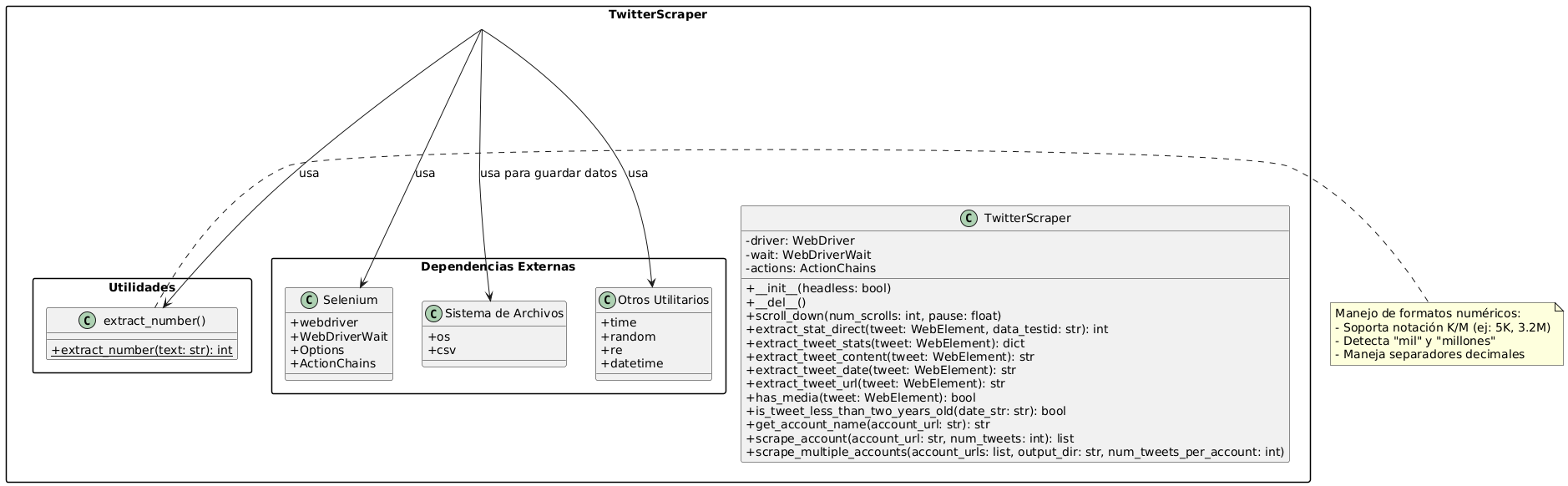
 Almacenar los datos en formato CSV para su posterior análisis

 Evitar mecanismos de detección de automatización implementados por la plataforma

# Arquitectura del Sistema

El sistema sigue una arquitectura modular basada en una única clase principal que encapsula toda la funcionalidad necesaria para la extracción de datos. Utiliza el patrón de diseño Facade para proporcionar una interfaz simplificada a las funcionalidades complejas de Selenium.

## Diagrama de Arquitectura



El sistema está compuesto por los siguientes componentes fundamentales:

 **Módulo principal**: Contiene la clase y las funciones auxiliares

 **Motor de automatización**: Implementado mediante Selenium WebDriver

 **Módulo de extracción**: Funciones especializadas para extraer diferentes tipos de datos

 **Sistema de almacenamiento**: Mecanismo para guardar los datos extraídos en formato CSV

# Componentes Principales

## Clase TwitterScraper

La clase TwitterScraper es el componente central del sistema y proporciona la siguiente funcionalidad:

 **Inicialización del navegador**: Configura el navegador Chrome con opciones específicas para evitar detección

 **Navegación**: Accede a las URLs de las cuentas objetivo

 **Extracción de datos**: Implementa métodos para obtener diferentes tipos de información

 **Procesamiento de datos**: Normaliza y estructura los datos extraídos

 **Almacenamiento**: Guarda los datos en archivos CSV

## Funciones Auxiliares

El sistema incluye funciones de soporte como

convert\_number\_text

formato numérico (ej. "10.2K", "5 mil") a valores numéricos enteros.

# Flujo de Trabajo

El flujo de trabajo del sistema sigue los siguientes pasos:

: Función que convierte textos con

1. **Inicialización**: Configuración del navegador con opciones anti-detección
2. **Iteración de cuentas**: Procesamiento secuencial de cada cuenta especificada

## Extracción por cuenta:

 Acceso a la URL de la cuenta

 Desplazamiento (scroll) para cargar publicaciones

 Identificación de elementos del DOM que contienen publicaciones  Filtrado de publicaciones por antigüedad

## Extracción de datos por publicación:

 Texto

 Fecha y hora  URL

 Estadísticas de interacción

 Detección de contenido multimedia

## Almacenamiento:

 Guardado de datos en archivos CSV específicos por cuenta  Generación de un archivo resumen de la extracción

# Funcionalidades Clave

## Extracción de Estadísticas

El sistema implementa múltiples estrategias para extraer estadísticas de interacción:

 Método primario: Búsqueda directa por atributos

 Método secundario: Análisis de elementos con roles específicos

 Método terciario: Análisis directo de elementos numéricos y su contexto visual

## Filtrado por Antigüedad

El sistema verifica la antigüedad de las publicaciones mediante:

 Extracción de atributos de elementos

 Conversión a objetos datetime de Python

 Comparación con la fecha actual para calcular la antigüedad en días  Filtrado de publicaciones con más de 730 días (2 años)

## Detección de Contenido Multimedia

La detección de imágenes y videos se realiza mediante:

 Búsqueda de selectores específicos relacionados con contenido multimedia  Detección de atributos relacionados con contenido visual

 Identificación de elementos con patrones específicos

# Mecanismos Anti-Detección

El sistema implementa las siguientes estrategias para evitar la detección por parte de la plataforma:

## Configuración del navegador:

 Desactivación de flags de automatización  Personalización del user-agent

 Desactivación de notificaciones y popups

## Comportamiento humanizado:

 Pausas aleatorias entre acciones  Desplazamientos naturales

 Visualización de elementos antes de extraer datos

## Manejo de popups:

 Detección y cierre de modales de inicio de sesión  Bypass de restricciones de visualización

# Manejo de Datos

## Estructura de Datos

Cada publicación extraída se estructura como un diccionario con los siguientes campos:

: Identificador de la cuenta

account\_id

: Contenido textual de la publicación

text

: Timestamp de publicación en formato ISO

timestamp

: URL única de la publicación

url

: Indicador booleano de presencia de contenido multimedia

has\_multimedia

: Número de comentarios/respuestas

comments

: Número de retweets

retweets

: Número de likes/me gusta

likes

: Número de veces compartido/guardado

shares

## Almacenamiento

Los datos se almacenan en dos tipos de archivos CSV:

 **Archivos por cuenta**: Contienen todas las publicaciones extraídas de una cuenta específica

 **Archivo resumen**: Contiene estadísticas generales de la extracción (cuentas, cantidad de publicaciones, timestamp)

# Limitaciones Conocidas

 El sistema depende de la estructura del DOM de la plataforma X, por lo que cambios en la interfaz pueden requerir actualizaciones

 La extracción de estadísticas puede ser imprecisa en algunos casos debido a la variabilidad en la presentación de datos

 La plataforma X implementa medidas contra web scraping que podrían bloquear la IP temporalmente si se realizan demasiadas extracciones consecutivas

 No se pueden extraer publicaciones que requieran inicio de sesión para ser visualizadas

# Requisitos Técnicos

## Dependencias

 Python 3.7+

 Selenium WebDriver

 Chrome/Chromium Browser

 ChromeDriver compatible con la versión del navegador

 Módulos estándar de Python: time, csv, re, random, os, datetime

## Entorno de Ejecución

 Sistema operativo compatible con Chrome/Chromium  Acceso a internet estable

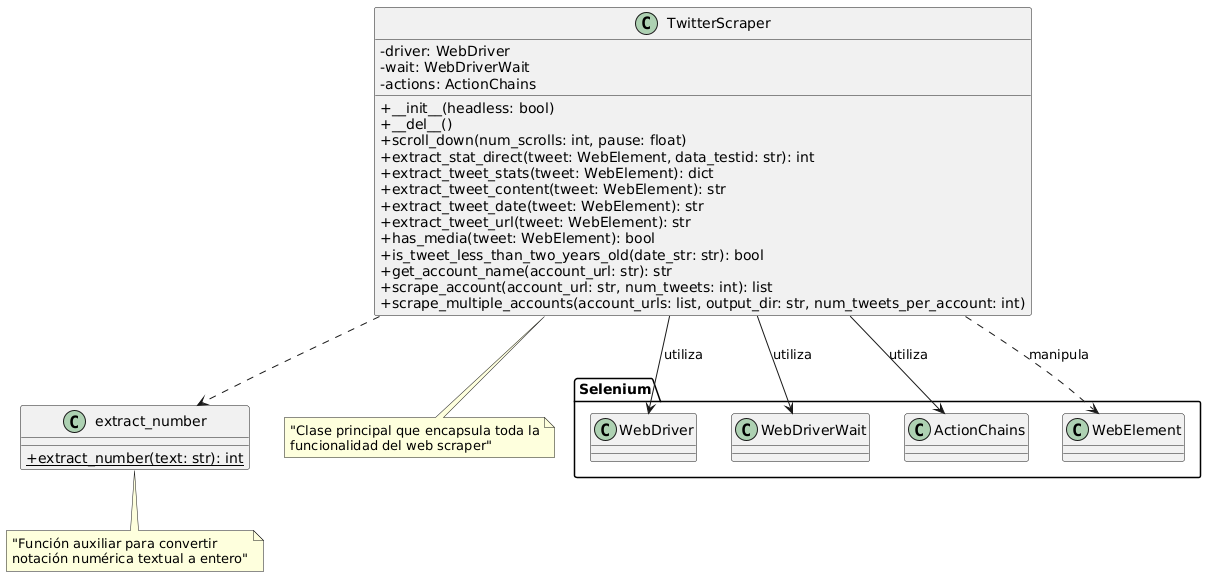
 Permisos de escritura en el sistema de archivos para guardar los CSV

## **Referencias**

1. IEEE Computer Society. (2008). *IEEE Standard for Software and System Test Documentation* (IEEE Std 829-2008). IEEE.
2. Selenium. (2023). *Selenium WebDriver Documentation*. <https://www.selenium.dev/documentation/webdriver/>
3. Python Software Foundation. (2023). *Python 3.7+ Documentation*. <https://docs.python.org/3/>
4. Google. (2023). *ChromeDriver - WebDriver for Chrome*. <https://chromedriver.chromium.org/>
5. Richardson, L. (2022). *Web Scraping with Python: Collecting Data from the Modern Web*. O'Reilly Media.
6. Twitter, Inc. (2023). *Platform X (Twitter) Developer Documentation*. <https://developer.twitter.com/en/docs>
7. Mitchell, R. (2022). *Web Scraping with Python: Collecting More Data from the Modern Web*. O'Reilly Media.
8. Mewburn, I., & Thomson, P. (2018). *The Good, the Bad and the Ugly of Web Scraping in Social Research*. Journal of Digital Social Research, 1(2), 22-34.

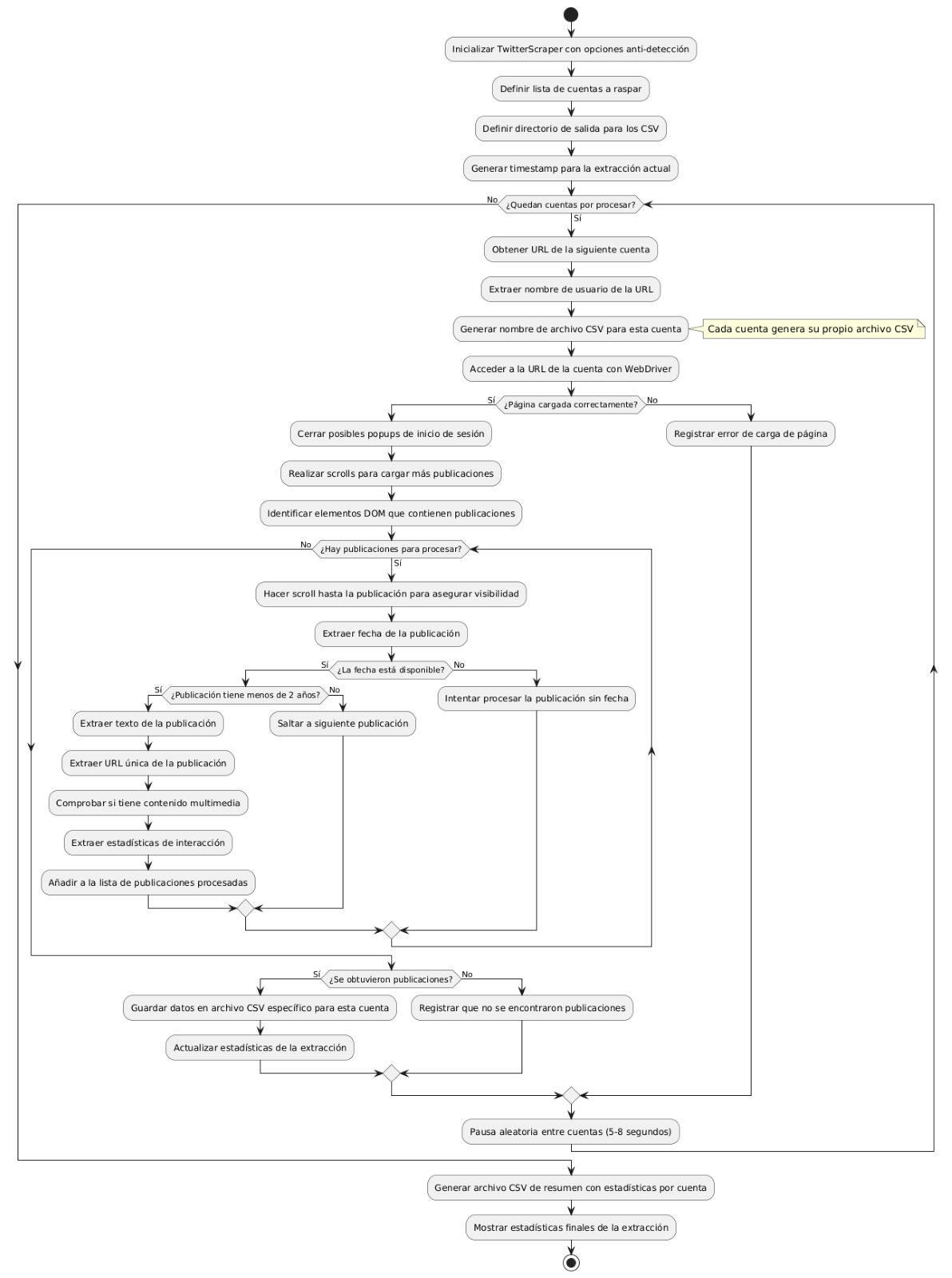
# Apéndice: Diagramas UML

## Diagrama de Clases



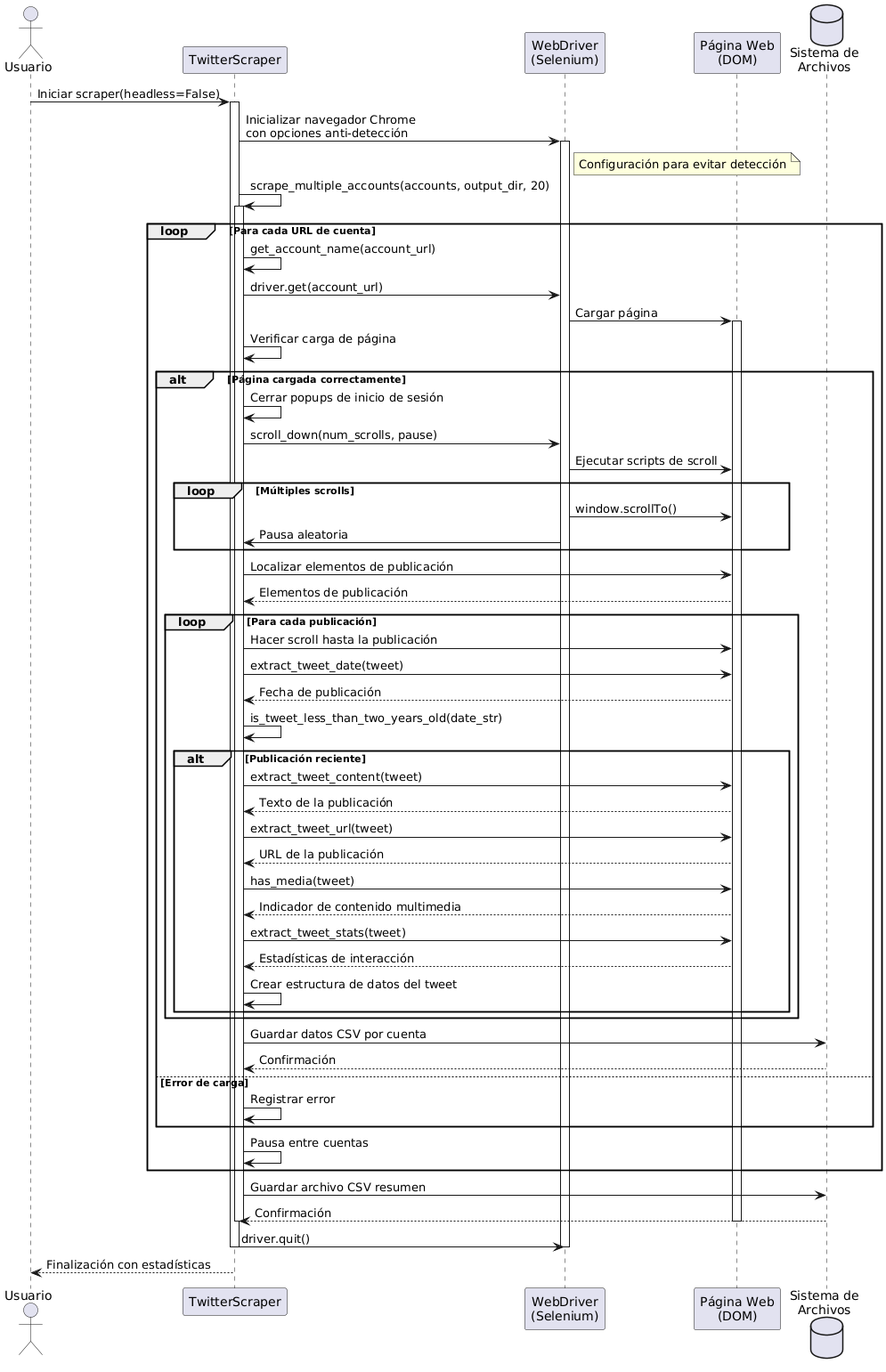
*Este diagrama muestra la estructura de clases del sistema, con la clase TwitterScraper como componente central que encapsula todas las funcionalidades principales, junto con sus métodos y propiedades.*

## Diagrama de Flujo



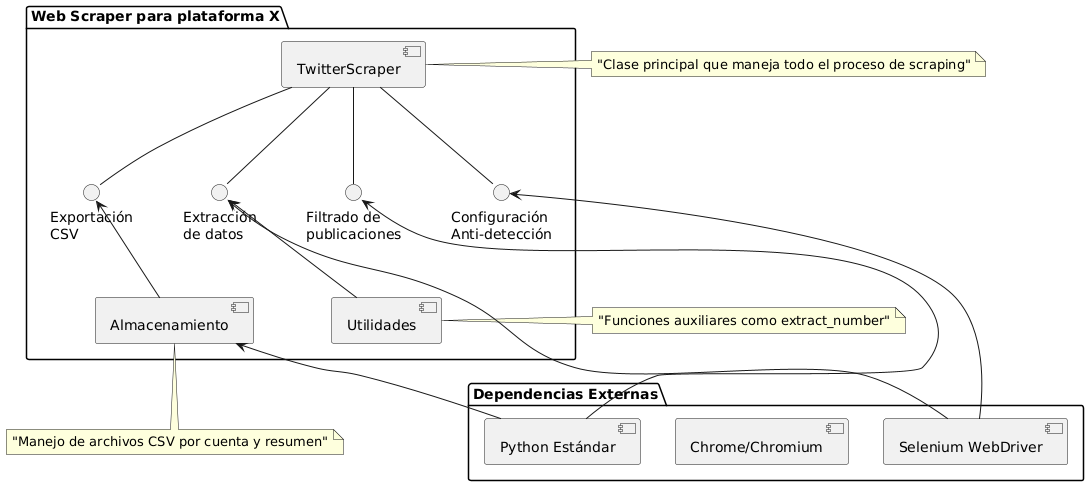
*Este diagrama ilustra el flujo completo de ejecución del sistema, desde la inicialización hasta el almacenamiento de datos, incluyendo todas las decisiones y procesos intermedios.*

## Diagrama de Secuencia



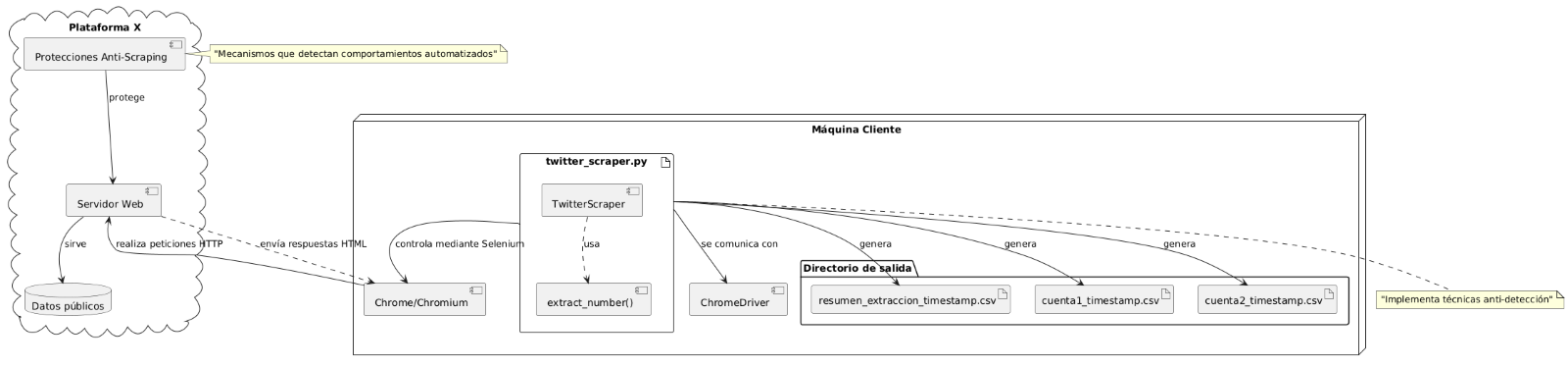
*Este diagrama representa la secuencia temporal de las interacciones entre los distintos componentes del sistema durante una ejecución típica, mostrando el orden de las llamadas y los mensajes entre objetos.*

## Diagrama de Componentes



*Este diagrama muestra los componentes principales del sistema y sus interrelaciones, identificando claramente las dependencias entre módulos y las interfaces que utilizan para comunicarse.*

## Diagrama de Despliegue



*Este diagrama representa la distribución física de los artefactos del sistema en la infraestructura de hardware, mostrando la relación entre los componentes de software y los nodos físicos donde se ejecutan.*